

Trails4health

|  |  |
| --- | --- |
| **Curso(s):** | Engenharia Informática |
| **Unidade(s) Curricular(es):** | Engenharia de Software  Programação para a Internet |
| **Ano Letivo:** | 2017/2018 |
| **Docente:** | Maria Clara Silveira  Noel Lopes |
| **Data:** | 24-11-2017 |

Índice

[1. Introdução 5](#_Toc503910195)

[2. Estado da arte 6](#_Toc503910196)

[2.1. Lista de casos de uso 6](#_Toc503910197)

[2.2. Tabela com comparação de funcionalidades 9](#_Toc503910198)

[2.3. Descrição de Padrões de Desenvolvimento 10](#_Toc503910199)

[SpiralDevelopment – Andreia Ernesto 10](#_Toc503910200)

[TwoTierView – Janilta Pires 11](#_Toc503910201)

[Quitting Time – Pedro Sanches 12](#_Toc503910202)

[3. Análise de Requisitos 16](#_Toc503910203)

[3.1. Descrição do tema do projeto 16](#_Toc503910204)

[3.2. Diagrama de Contexto 17](#_Toc503910205)

[3.3. Tabela de Atores, objetivos e respetivos Casos de Uso 18](#_Toc503910206)

[3.4. Diagrama de Casos de Uso 19](#_Toc503910207)

[3.5. Descrição dos Casos de Uso 20](#_Toc503910208)

[3.6. Diagramas de Sequência 32](#_Toc503910209)

[3.7. Diagramas de Classes completo 51](#_Toc503910210)

[3.8. Diagrama de Estados 52](#_Toc503910211)

[3.9. Diagrama de atividades 53](#_Toc503910212)

[3.10. Diagrama de componentes 54](#_Toc503910213)

[3.11. Diagrama de pacotes 54](#_Toc503910214)

[3.12. Diagrama de instalação 55](#_Toc503910215)

[3.13. Semântica de Classes 56](#_Toc503910216)

[Classe Turista: 57](#_Toc503910217)

[Classe Agenda\_Turista\_Trilho: 60](#_Toc503910218)

[Classe Trilho: 64](#_Toc503910219)

[1. Protótipo 66](#_Toc503910220)

[1.1. Página inicial 66](#_Toc503910221)

[1.2. Página de registo 67](#_Toc503910222)

[1.3. Página após registo 69](#_Toc503910223)

[1.4. Página Login 70](#_Toc503910224)

[1.5. Página Agendamentos 71](#_Toc503910225)

[1.6. Página Novo agendamento 72](#_Toc503910226)

[Conclusão 73](#_Toc503910227)

[Anexos](#_Toc503910228)

Índice de Figuras

[Figura 1 - Diagrama de contexto 17](file:///D:\Admin\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc503910167)

[Figura 2 - Diagrama de casos de uso, alguns fora da fronteira 19](file:///D:\Admin\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc503910168)

[Figura 3 - Diagrama de casos de uso, apenas casos de uso dentro da fronteira 20](file:///D:\Admin\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc503910169)

[Figura 4 - Diagrama de sequência "Inserir professor" 32](#_Toc503910170)

[Figura 5 - Diagrama de sequência “Consultar professor” 33](#_Toc503910171)

[Figura 6 - Diagrama de sequência “Editar professor” 34](#_Toc503910172)

[Figura 7 - Diagrama de sequência “Eliminar professor” 35](#_Toc503910173)

[Figura 8 - Diagrama de sequência “Registar turista” 36](#_Toc503910174)

[Figura 9 - Diagrama de sequência “Consultar perfil” 37](#_Toc503910175)

[Figura 10 - Diagrama de sequência “Editar perfil” 38](#_Toc503910176)

[Figura 11 - Diagrama de sequência “Desativar perfil” 39](#_Toc503910177)

[Figura 12 - Diagrama de sequência “Consultar histórico de trilhos percorridos” 40](file:///D:\Admin\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc503910178)

[Figura 13 - Diagrama de sequência “Autenticar utilizador” 41](file:///D:\Admin\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc503910179)

[Figura 14 - Diagrama de sequência “Consultar aptidão física” 42](file:///D:\Admin\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc503910180)

[Figura 15 - Diagrama de sequência “Configurar trilho” 43](#_Toc503910181)

[Figura 16 - Diagrama de sequência “Consultar utilizadores que percorreram determinado trilho” 44](file:///D:\Admin\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc503910182)

[Figura 17 - Diagrama de sequência “Agendar trilho” 45](#_Toc503910183)

[Figura 18 - Diagrama de sequência “Alterar trilho” 46](#_Toc503910184)

[Figura 19 - Diagrama de sequência “Consultar trilho agendado” 47](#_Toc503910185)

[Figura 20 - Diagrama de sequência “Cancelar trilho” 48](#_Toc503910186)

[Figura 21 - Diagrama de sequência "Atualizar estado de agendamento do trilho" 49](#_Toc503910187)

[Figura 22 - Diagrama de sequência "Atualizar tempo gasto" 50](#_Toc503910188)

[Figura 23 - Diagrama de classes 51](file:///D:\Admin\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc503910189)

[Figura 24 - Diagrama do estado “Agendar Trilho” 52](file:///D:\Admin\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc503910190)

[Figura 25 - Diagrama de atividades "Agendar trilho" 53](#_Toc503910191)

[Figura 26 - Diagrama de componentes Trails4Health 54](file:///D:\Admin\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc503910192)

[Figura 27 - Diagrama de pacotes Trails4Health 54](#_Toc503910193)

[Figura 28 - Diagrama de instalação 55](file:///D:\Admin\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc503910194)

1. Introdução

Na era moderna somos invadidos por uma tecnologia megalómana que nos leva para além daquilo que somos. A originalidade nos dias de hoje é algo evidente por todo o lado, toda a gente é original, pois todos tentamos ser diferentes, porém acabamos por ser todos iguais. Se um olhar para as realidades ajuda a perceber aquilo que e quem somos, um olhar contrariamente a isso ajuda-nos a ser diferentes e aí sim, originais!

Este projeto advém daí, o facto da importância do Ser original, inovador, aquele que tentamos e queremos ser. E com este projeto pretendemos delinear um futuro, um futuro que nós enquanto alunos e para a empresa que tentaremos ajudar.

Desta forma, é importante garantir que haja um negócio que viabilize o desenvolvimento de um determinado produto. Sabendo que existe uma grande necessidade de desenvolver um *software* para dinamizar uma empresa com a finalidade de melhorar os serviços prestados aos clientes e permitindo uma maior facilidade de acesso.

“Trails4Health” é uma empresa que pretende auxiliar os amantes de trilhos pedestres na zona da Serra da Estrela, de modo a dar a conhecer aos turistas o melhor que existe na região como a gastronomia, a fauna, a flora e as atividades praticadas. O projeto visa desenvolver uma aplicação criada pelos estudantes do Instituto Politécnico da Guarda auxiliado ao método de trabalho existente nas grandes empresas, de maneira a ter uma ideia do que o futuro reserva.

1. 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 | ✔ | ✔ |

# 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

O projeto apresentado aos alunos de Engenharia Informática do 3º ano consiste na criação de uma aplicação com a finalidade de dinamizar a região histórica da Serra da Estrela e promover o turismo. O projeto será desenvolvido em UML e a aplicação será concebida na linguagem de programação ASP.NET CORE MVC. A caraterística principal da aplicação é o agendamento, onde o turista pode marcar a sua visita a um dos trilhos apresentados. Com isto, basta o utilizador registar-se na aplicação, escolher o trilho pedestre que pretende percorrer e quando o deseja fazer. Assim, os utilizadores na comodidade do seu lar, conseguirão ver as caraterísticas de cada trilho e o que poderão encontrar ao percorrê-lo.

Porém, outra caraterística importante foi a necessidade na criação de privilégios por parte do utilizador e professor de desporto. O professor de desporto terá mais privilégios em relação à aplicação do que o turista, na medida em que quando o *login* é efetuado, o professor de desporto pode adicionar, editar, consultar e eliminar um trilho enquanto o turista apenas poderá consultar.

# Diagrama de Contexto

O diagrama de contexto é uma ferramenta para modelar o objetivo de um projeto através de um diagrama.

No caso deste projeto o diagrama de contexto é o apresentado na Figura 1.

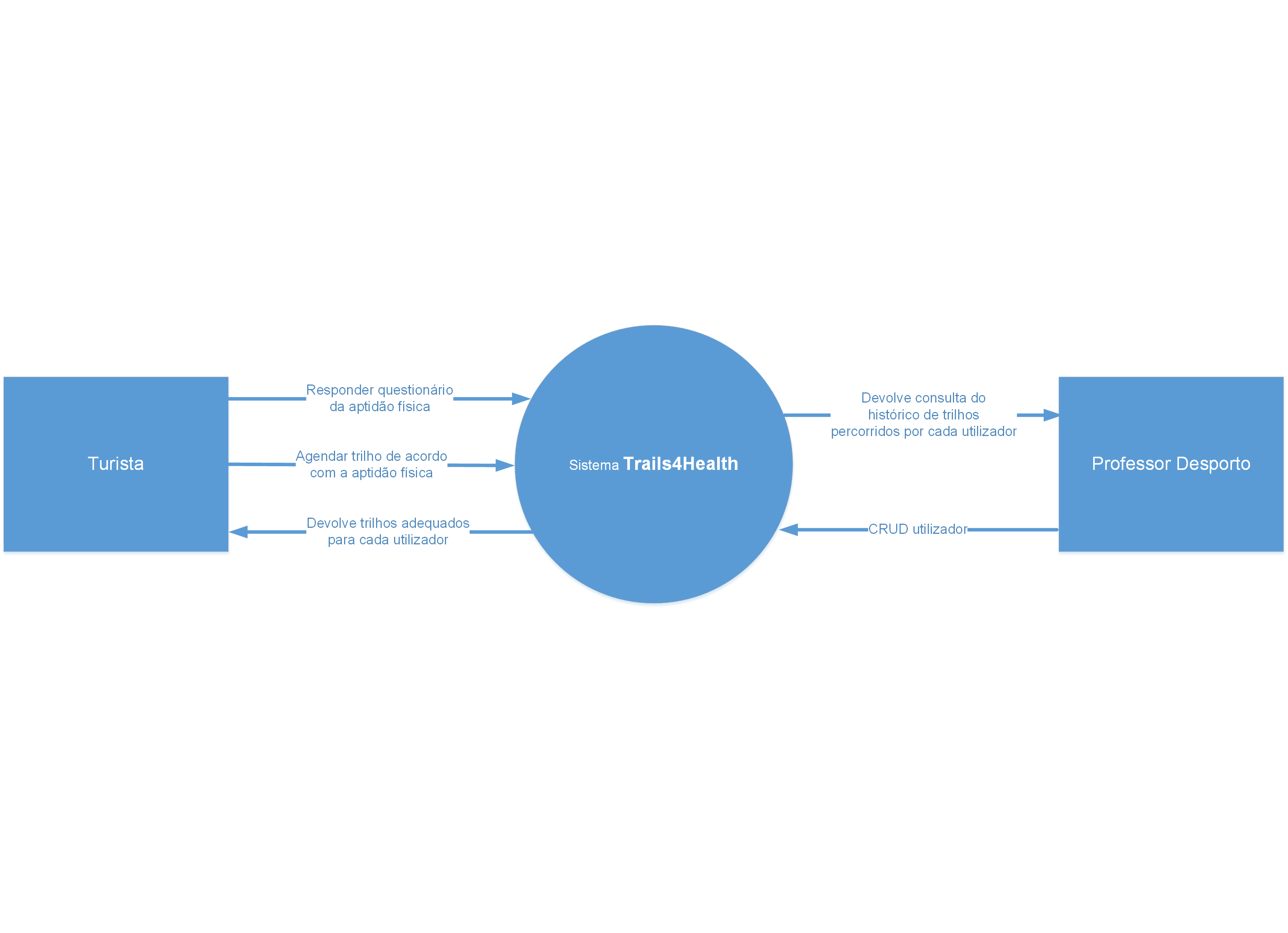


Figura 1 - Diagrama de contexto

# Tabela de Atores, objetivos e respetivos Casos de Uso

|  |  |  |
| --- | --- | --- |
| **Possíveis Atores** | **Nome do caso de uso** | **Descrição do caso de uso** |
| **Turistas** | Configurar Trilho | Este caso de uso tem como objetivo apresentar os melhores trilhos consoante a aptidão física de cada utilizador. |
| Alterar a data prevista de inicio do trilho | Este caso de uso tem como objetivo o Turista alterar a data prevista do inicio do trilho. |
| Consultar trilhos agendados | Este caso de uso tem como objetivo o Turista consulta o seu trilho agendado. |
| Cancelar trilho agendado | O objetivo é o turista fazer o cancelamento do trilho agendado. |
| Consultar aptidão física | Este caso de uso tem como objetivo o turista consultar a sua aptidão física. |
| Realizar trilho | Este caso de uso tem como objetivo o turista confirmar se realizou o trilho no final de o percorrer. |
| **Professor de Desporto** | Consultar utilizadores que fizeram o trilho | O objetivo deste caso de uso é permitir ao professor de desporto visualizar os utilizadores que já percorreram determinado trilho. |
| Consultar histórico trilho percorrido | Este caso de uso permitir ao professor de desporto consultar o histórico de trilhos percorridos por cada utilizador. |
| Inserir utilizador | O objetivo deste caso de uso é o ator aceder ao sistema com o intuito de criar um utilizadores |
| Consultar utilizador | O objetivo é o professor de desporto puder ver todos os turistas e o turista puder consultar o seu próprio perfil. |
| Editar utilizador | O objetivo é o professor de desporto puder ver editar os turistas e o turista puder editar o seu próprio perfil. |
| Eliminar utilizador | O objetivo é o professor de desporto puder ver eliminar os turistas e o turista puder eliminar o seu próprio perfil. |
| Cancelar trilho agendado | O objetivo é o professor de desporto cancelar o trilho agendado caso não haja as condições necessárias de o percorrer. |

# Diagrama de Casos de Uso

**Com casos de uso fora da fronteira**

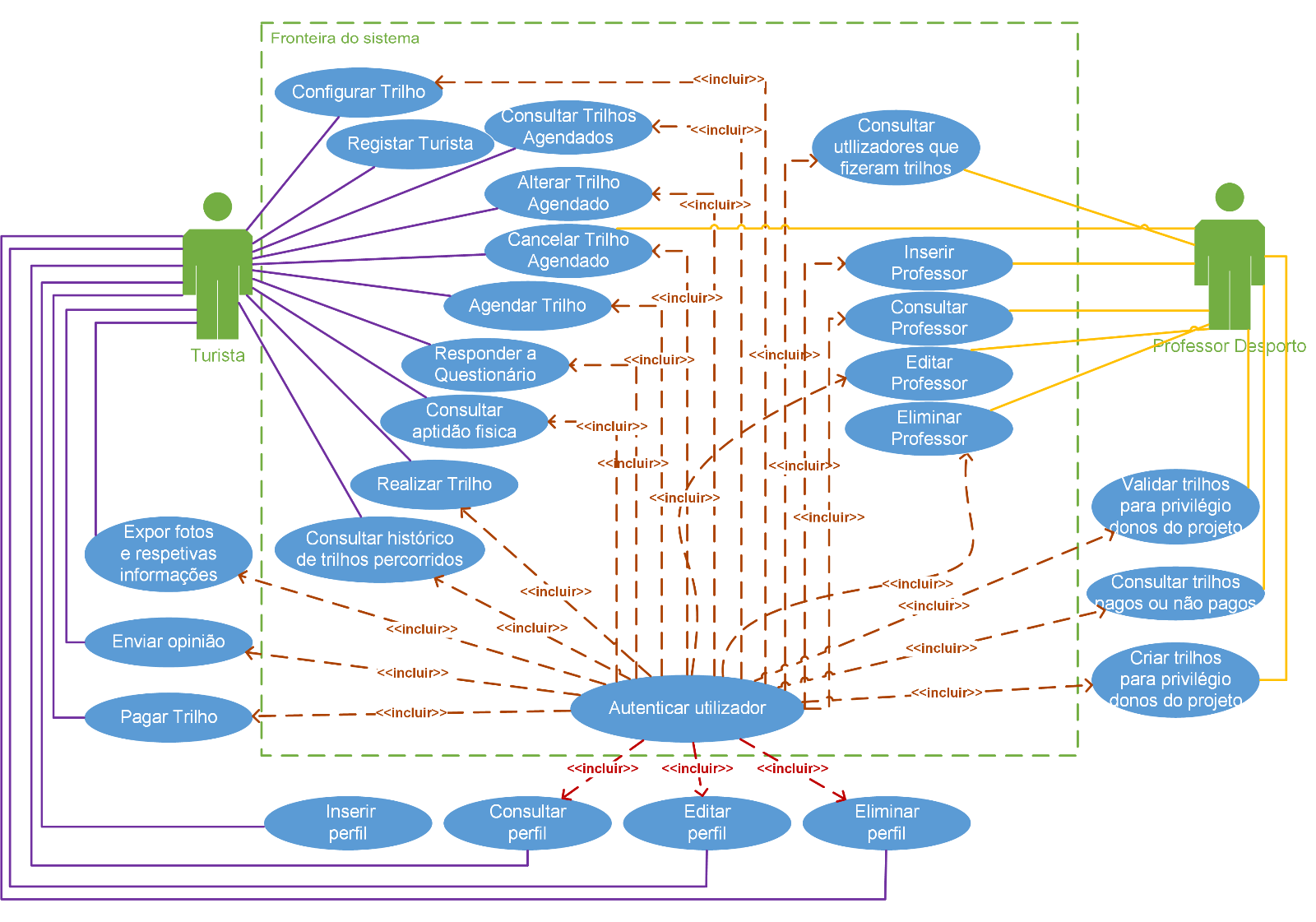


Figura 2 - Diagrama de casos de uso, alguns fora da fronteira

**Apenas casos de uso dentro da fronteira**

Os casos de uso referentes ao desenvolvimento dos objetivos deste projeto é o apresentado na figura abaixo.

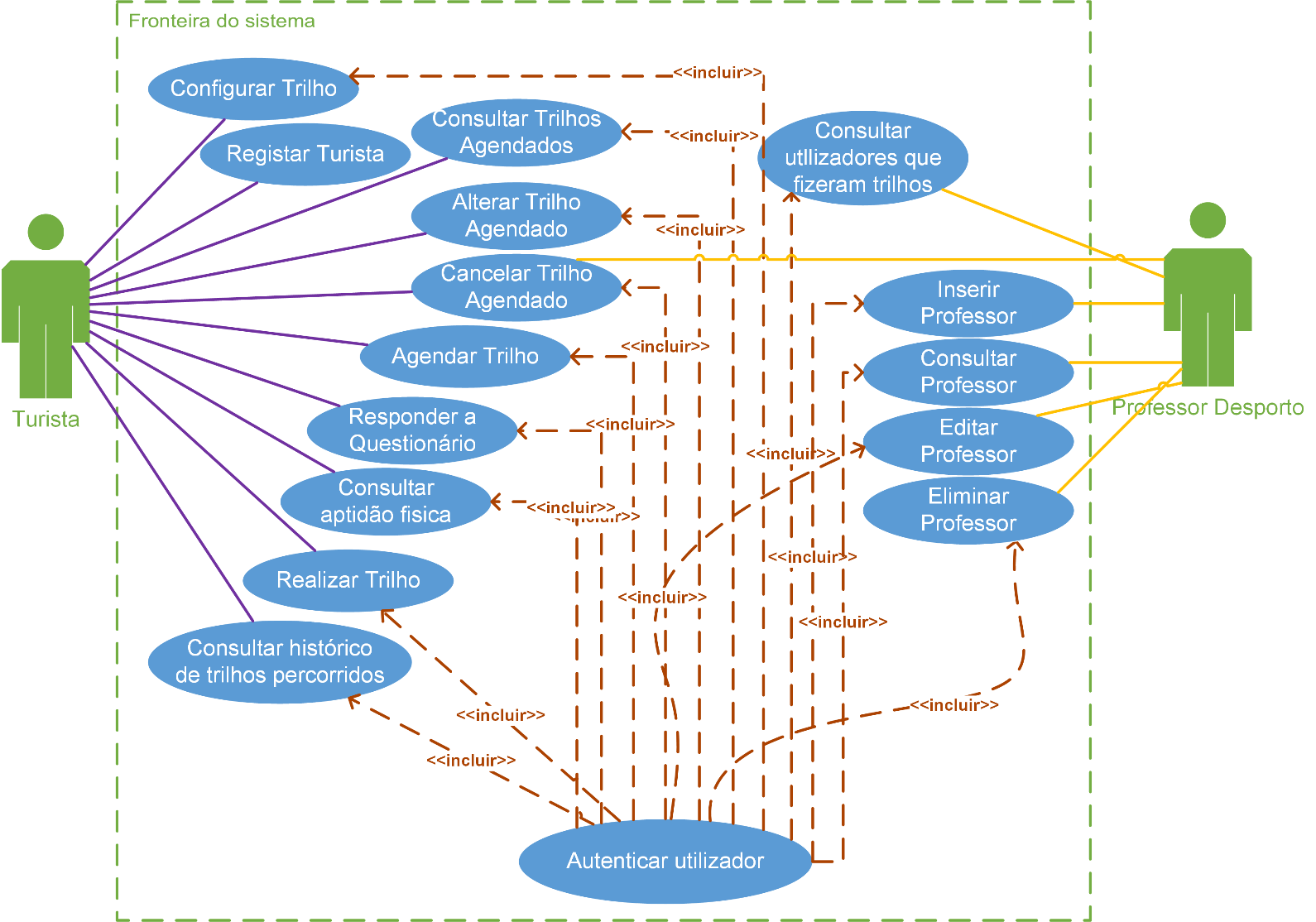


Figura 3 - Diagrama de casos de uso, apenas casos de uso dentro da fronteira

# Descrição dos Casos de Uso

Nesta seção faz-se uma descrição de cada caso de uso que está dentro do sistema da fronteira, com o objetivo de um melhor entendimento entre programador do projeto e o utilizador.

O campo tamanho é uma métrica que subentende o trabalho e complexidade do desenvolvimento de cada caso de uso. Usamos a métrica de Fibonacci (1, 2, 3, 5, 7, 13, 21…).

**Inserir professor**

|  |  |
| --- | --- |
| **Nome:** | Inserir professor |
| **Descrição:** | O objetivo deste caso de uso é o professor aceder ao sistema através da sua conta de administrador com o intuito de criar outros professores |
| **Tamanho:** | 13 |
| **Pré-condição:** | Não têm |
| **Caminho principal:** | 1. O utilizador seleciona a opção de “Inserir Professor” 2. O sistema disponibiliza o formulário de inserção de professor com todos os campos a preencher 3. O ator preenche o formulário de inserção de professor (introduz o Nome, Password, Morada...) 4. O sistema pede para confirmar 5. O professor 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 professor**

|  |  |
| --- | --- |
| **Nome:** | Consultar professor |
| **Descrição:** | O objetivo é o professor de desporto puder ver todos os professores. |
| **Tamanho:** | 7 |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O professor de desporto seleciona a opção “Consultar professores” 2. O sistema apresenta uma interface “Consultar professor” com um formulário para pesquisa e uma tabela com todos os turistas. 3. O ator vai pesquisar um professor por Nome ou NIF. 4. O sistema mostra o resultado da pesquisa feita. |
| **Caminho alternativo:** | 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 professor**

|  |  |
| --- | --- |
| **Nome:** | Editar professor |
| **Descrição:** | O objetivo é o professor de desporto puder editar os professores. |
| **Tamanho:** | 7 |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O professor de desporto seleciona a opção “Editar professor” 2. O sistema apresenta uma interface “Editar professor” com um formulário para pesquisa e uma tabela com todos os professores. 3. O ator vai pesquisar um professor 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.  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 professor**

|  |  |
| --- | --- |
| **Nome:** | Eliminar professor |
| **Descrição:** | O objetivo é o professor de desporto puder eliminar um professor |
| **Tamanho:** | 7 |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O professor de desporto seleciona a opção “Eliminar professor” 2. O sistema apresenta uma interface “Eliminar professor” com um formulário para pesquisa e uma tabela com todos os professores. 3. O ator vai pesquisar um professor 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** |

**Registar turista**

|  |  |
| --- | --- |
| **Nome:** | Registar turista |
| **Descrição:** | O turista regista-se, através de um formulário de registar utilizador. |
| **Tamanho:** | 21 |
| **Pré-condição:** |  |
| **Caminho principal:** | 1. O ator seleciona a opção “Registar”; 2. O sistema apresenta os campos para a registo do utilizador (Nome/E-mail, Password, confirmação da password, morada, NIF…); 3. O ator preenche os campos e valida os dados; 4. O sistema verifica se os campos foram devidamente preenchidos e executa o registo com sucesso; |
| **Caminho alternativo:** | 3. a) O utilizador não preencheu devidamente os campos. |
| **Suplementos ou adornos:** | Testar se os campos obrigatórios estão preenchidos.  Testar o campo do NIF através do algoritmo do dígito de controlo.   1. Garantir que o sistema funciona mesmo quando existe excesso de procura |
| **Pós-Condição:** | Não existe. |

**Autenticar utilizador**

|  |  |
| --- | --- |
| **Nome:** | Autenticar Utilizador |
| **Descrição:** | O turista/professor de desporto ao autenticarem-se, através da conta previamente registado, conseguem aceder a informações privilegiadas através do seu perfil. |
| **Tamanho:** | 21 |
| **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. |
| **Tamanho:** | 7 |
| **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. |
| **Tamanho:** | 21 |
| **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 |
| **Tamanho:** | 7 |
| **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. |
| **Tamanho:** | 21 |
| **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. |
| **Tamanho:** | 21 |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O ator seleciona a opção “Agendar Trilho”; 2. O sistema apresenta os trilhos já Agendados anteriormente; 3. Apresenta a opção “Agendar novo Trilho” 4. O ator determina a data prevista que pretende percorrer, o local de inicio fim, e o tempo que pretende gastar em cada percurso; 5. 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); 6. O ator seleciona o trilho que pretende percorrer de acordo com as propostas apresentadas; 7. 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 |
| **Tamanho:** | 13 |
| **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 |
| **Tamanho:** | 21 |
| **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 |
| **Tamanho:** | 13 |
| **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

Os **diagramas de sequência** mostram as interações entre objetos segundo uma perspetiva temporal.

Os diagramas apresentados no seguimento deste relatório ilustram a perspetiva temporal de todos os casos de uso desenhados dentro da fronteira.

**Inserir Professor**



Figura 4 - Diagrama de sequência "Inserir professor"

**Consultar Professor**



Figura 5 - Diagrama de sequência “Consultar professor”

**Editar Professor**



Figura 6 - Diagrama de sequência “Editar professor”

**Eliminar Professor**



Figura 7 - Diagrama de sequência “Eliminar professor”

**Registar Turista**



Figura 8 - Diagrama de sequência “Registar turista”

**Consultar perfil**



Figura 9 - Diagrama de sequência “Consultar perfil”

**Editar perfil**



Figura 10 - Diagrama de sequência “Editar perfil”

**Desativar perfil**



Figura 11 - Diagrama de sequência “Desativar perfil”

**Consultar histórico de trilhos percorridos**



Figura 12 - Diagrama de sequência “Consultar histórico de trilhos percorridos”

**Autenticar Utilizador**

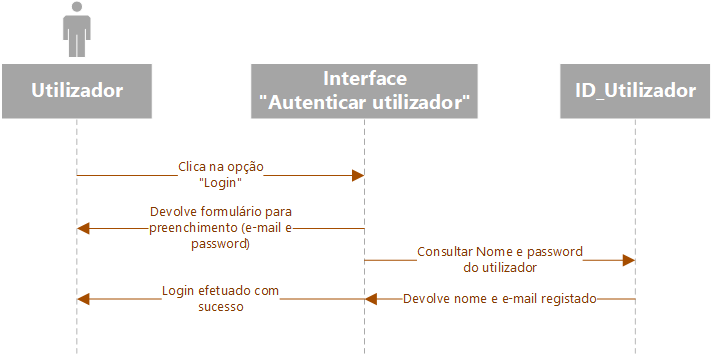


Figura 13 - Diagrama de sequência “Autenticar utilizador”

**Consultar Aptidão Física**

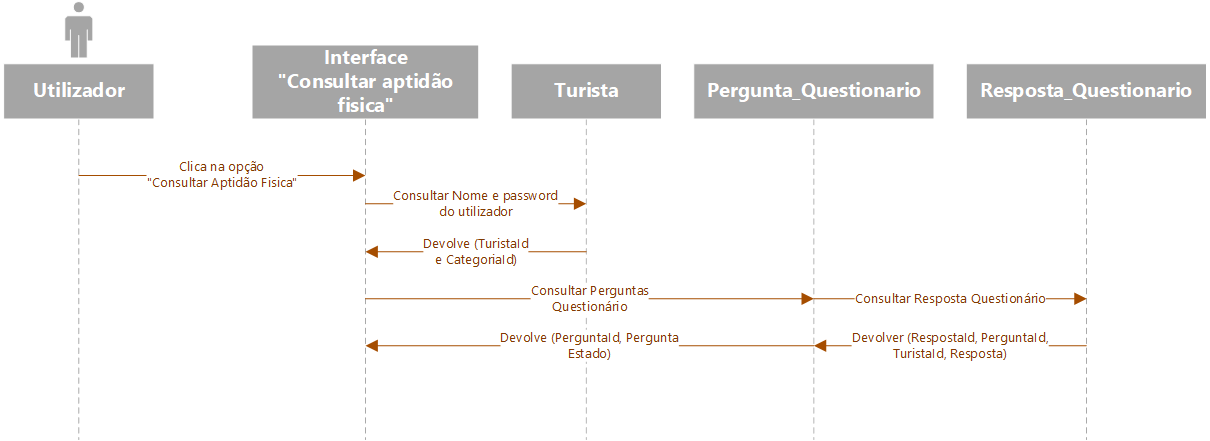


Figura 14 - Diagrama de sequência “Consultar aptidão física”

**Configurar Trilho**



Figura 15 - Diagrama de sequência “Configurar trilho”

**Consultar utilizadores que percorreram determinado trilho**



Figura 16 - Diagrama de sequência “Consultar utilizadores que percorreram determinado trilho”

**Agendar Trilho**



Figura 17 - Diagrama de sequência “Agendar trilho”



**Alterar Agendamento do Trilho**

Figura 18 - Diagrama de sequência “Alterar trilho”

**Consultar Trilho Agendado**



Figura 19 - Diagrama de sequência “Consultar trilho agendado”

**Cancelar Trilho Agendado**



Figura 20 - Diagrama de sequência “Cancelar trilho”

**Atualizar Estado Agendamento do Trilho**



Figura 21 - Diagrama de sequência "Atualizar estado de agendamento do trilho"

**Atualizar Tempo Gasto**



Figura 22 - Diagrama de sequência "Atualizar tempo gasto"

# Diagramas de Classes completo

Nesta seção deste relatório pode ver-se o diagrama de classes da aplicação, na figura 17. Este tipo de diagrama demonstra o relacionamento entre as várias classes. Cada classe é constituída pelo nome, atributos e palas operações que irá executar.

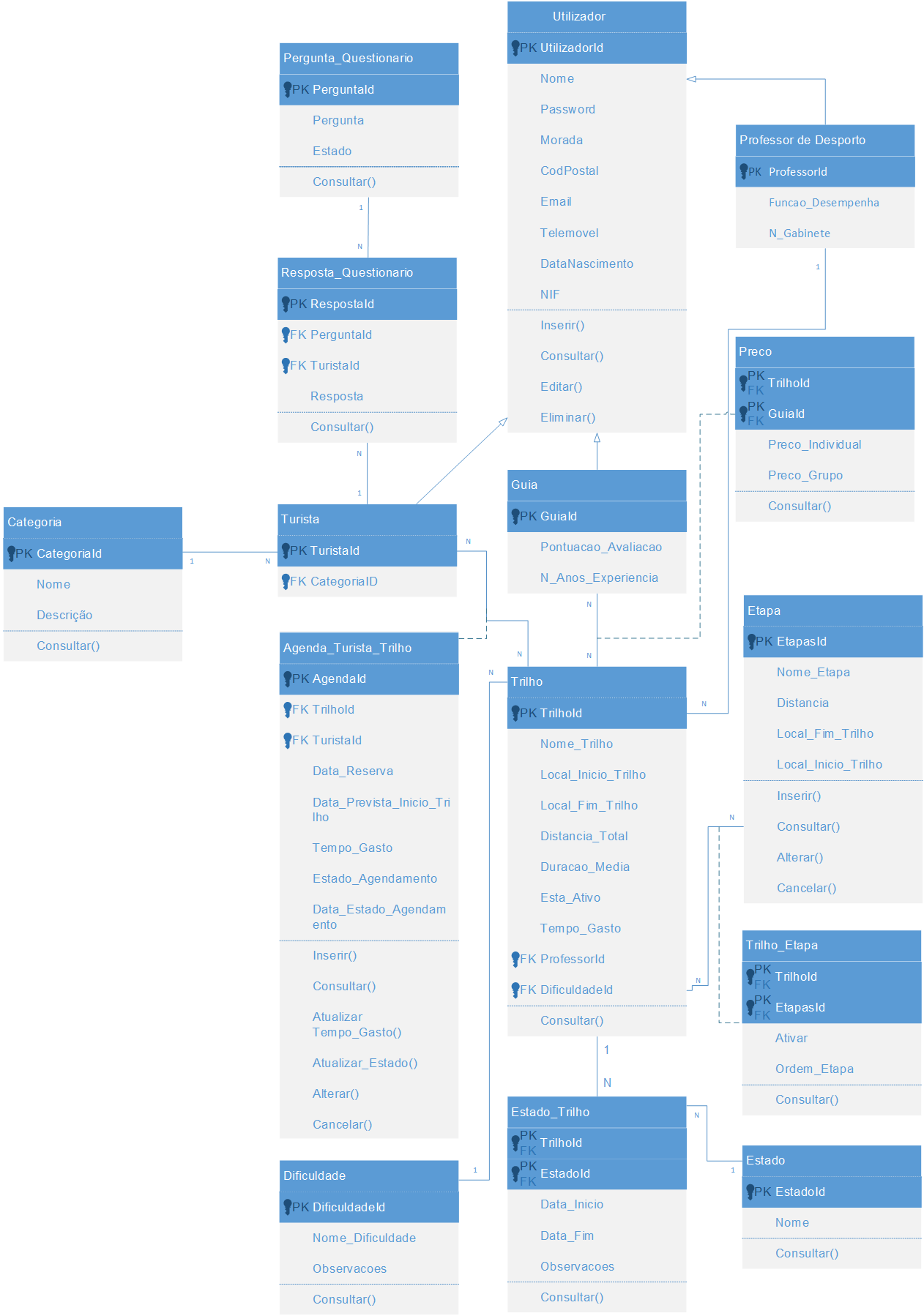


Figura 23 - Diagrama de classes

# Diagrama de Estados

Na figura abaixo podemos ver o diagrama de estados do caso de uso “Agendar Trilho” que mostra a transição de estado de um agendamento.

**Agendar Trilho**

O agendamento inicialmente tem o estado “Agendado”, se a data de inicio passar e o estado não tiver sido alterado então o estado mudará para “Não realizado”. Além disso existe também o estado “Realizado” e “Cancelado”. O estado cancelado significa que faltaram condições para percorrer o trilho, como por exemplo condições meteorológicas.



Figura 24 - Diagrama do estado “Agendar Trilho”

# Diagrama de atividades

O diagrama de atividades representa os ﬂuxos operacionais do sistema descrevendo assim de uma forma genérica e organizada as operações que constituem a aplicação.

O diagrama de atividades da classe “Agendar trilho” pode consultar-se na figura abaixo.

Uma imagem contendo texto, sinal

Descrição gerada com muito alta confiança

Figura 25 - Diagrama de atividades "Agendar trilho"

# 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.

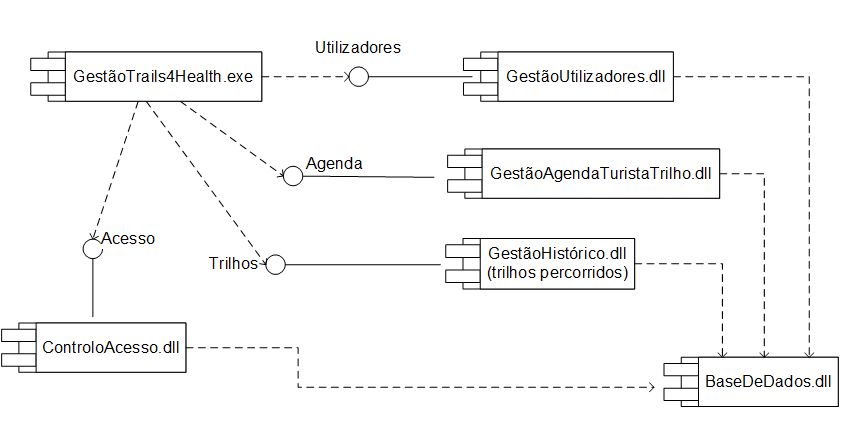


Figura 26 - Diagrama de componentes Trails4Health

# Diagrama de pacotes



Figura 27 - Diagrama de pacotes Trails4Health

# 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.

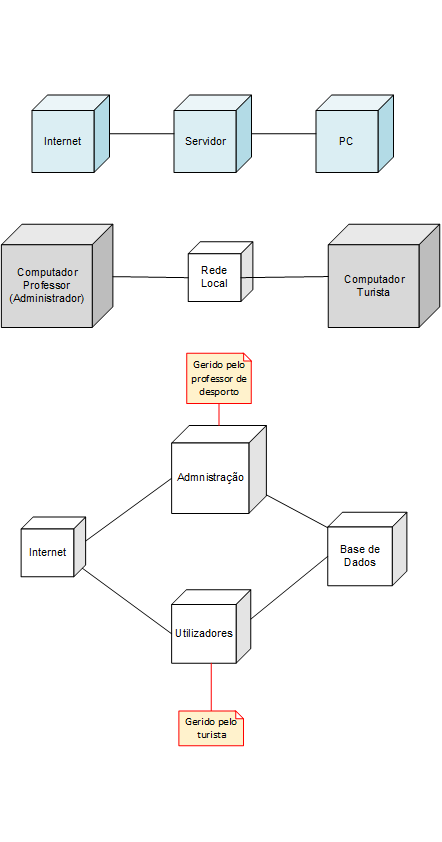


Figura 28 - Diagrama de instalação

# Semântica de Classes

**Algoritmo do dígito de controlo do NIF para classe de turista e professor**

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: Turista | | | | | |
| Nome do Campo | Tipo de dados | Descrição | Valores Válidos | Formato | Restrições |
| TuristaId | Numeração automática | Número sequencial que identifica univocamente a cada turista | Maior que 0 | Até 6 digitos | Gerado pelo sistema/Não alterável |
| Nome | String | Nome do turista | Letras | Até 30 caracteres | Obrigatório |
| Password | String | Password para fazer login | - | Constituído por uma maiúscula, números e pelo menos um caracter especial. | Tem de ter |
| Morada | String | Morada do turista | - |  | Obrigatório |
| Cod\_Postal | Número | Código Postal referente a morada | - | - | Obrigatório |
| Email | String | Email do turista | - | algo@algo.algo | Obrigatório |
| Telemovel | Number | Telemóvel do turista | - | - | Obrigatório |
| DataNascimento | Data | Data de nascimento do turista | - | Menor que a data do sistema | Obrigatório |
| NIF |  |  |  | Validação pelo digito de controlo | Obrigatório |
| CategoriaId | Número | Indica o Id da categoria em que o turista se insere |  | Botão opção ou equivalente | - |

**Dicionário de dados:**

**Descrição das Operações:**

Registar Turista

|  |  |
| --- | --- |
| Operações | |
| Nome | **Descrição** |
| Registar turista | Operação que permite criar um novo utilizador do tipo turista   1. O sistema gera o TuristaId 2. Introduzir nome 3. Introduzir password 4. Introduzir Morada 5. Introduzir Cod\_Postal 6. Introduzir Email 7. Introduzir Telemovel 8. Introduzir DataNascimento 9. Introduzir NIF com validação Dígito de controlo (chama algoritmo valida) 10. Selecionar a categoria 11. Criar novo registo |

Consultar perfil

|  |  |
| --- | --- |
| Operações | |
| Nome | **Descrição** |
| Consultar perfil | 1. Permite a consulta dos dados do turista em questão através do seu Id |

Editar perfil

|  |  |
| --- | --- |
| Operações | |
| Nome | **Descrição** |
| Editar perfil | Operação que permite alterar os dados do turista:   1. Alterar nome 2. Alterar password 3. Alterar Morada 4. Alterar Cod\_Postal 5. Alterar Email 6. Alterar Telemovel 7. Alterar DataNascimento   … |

Desativar Turista

|  |  |
| --- | --- |
| Operações | |
| Nome | **Descrição** |
| Desativar Turista | 1. Permite a desativação da do turista em questão através do seu Id. |

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

1. Registar Turista
2. Consultar perfil
3. Editar perfil
4. Desativar perfil
5. Consultar histórico de trilhos percorridos
6. Autenticar utilizador
7. Consultar aptidão física
8. Configurar trilho
9. Consultar utilizadores que percorreram determinado trilho
10. Agendar Trilho
11. Alterar Trilho Agendado
12. Consultar Trilho Agendado
13. Cancelar Trilho Agendado

## 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 lista do trilho | Obrigatório/não alterável |
| TuristaId(FK) | Numerico | Indica o ID de um turista | Maior que 0 | Até 7 Caracteres | 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 | Não pode ser uma letra | AAAA/MM/DD | Obrigatório/Alterável/  Preenchido pelo utilizador |
| Tempo\_Gasto | Numerico | Tempo que foi gasto durante o percurso | --- | HH:MM:SS | Preenchido |
| Estado\_Agendamento | String | Indica o estado do agendamento do trilho se foi (realizado, cancelado…) | -------- | ----- | Alterável |
| Data\_Estado\_Agendamento | Data Time | Indica o dia como está o agendamento do trilho | Não pode ser uma letra | AAAA/MM/DD | Alterável |

**Dicionário de dados:**

**Descrição das Operações:**

**Classe** Agenda\_Turista­\_Trilho -AgendarTrilho

|  |  |
| --- | --- |
| **Nome** | **Descrição** |
| Criar()// **Get** | **Operação que permite mostrar interface para criar um novo agendamento**   1. Devolve a View da criação do agendamento do trilho |
| Criar()// **Post** | Operação que permite criar um novo Agendamento   1. O sistema mostra a lista de todos os trilhos 2. Seleciona umTrilho 3. Seleciona a Data prevista inicio do trilho 4. Cria nova agenda |

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

|  |  |
| --- | --- |
| **Nome** | **Descrição** |
| Criar()// **Get** | **Operação que permite mostrar interface para criar um novo agendamento**   1. Devolve a View da Alteração agendamento do trilho |
| Criar()// **Post** | Operação que permite Alterar os dados do trilho agendado   1. O sistema mostra a lista de todos os trilhos agendados 2. Seleciona umTrilho 3. Altera a data prevista do inicio do trilho 4. Altera a data da reserva do agendamento 5. Atualiza os dados |

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

|  |  |
| --- | --- |
| **Nome** | **Descrição** |
| Criar()// **Get** | **Operação que permite mostrar interface para consultar o trilho agendado**   1. Devolve a View da Consultar trilho agendado |
| Criar()// **Post** | Operação que permite Consultar os dados do trilho agendado   1. O sistema mostra a lista de todos os trilhos agendados 2. Consulta o trilho 3. Consulta o turista que fez o agendamento do trilho 4. Consulta a dada que foi agendado o trilho |

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

|  |  |
| --- | --- |
| **Nome** | **Descrição** |
| Criar()// **Get** | **Operação que permite mostrar interface para Cancelar o trilho agendado**  Devolve a View da Cancelar trilho agendado |
| Criar()// **Post** | Operação que permite Cancelar os dados do trilho agendado   1. O sistema mostra a lista de todos os trilhos agendados 2. Consulta o trilho 3. Consulta o turista que fez o agendamento do trilho 4. Cancela a data da reserva 5. Cancela data prevista inicio do trilho 6. Atualiza os dados |

## Classe Trilho:

**Dicionário de dados: Consultar Trilhos**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Classe: Trilho | | | | | |
| Nome do Campo | Tipo de dados | Descrição | Valores Válidos | Formato | Restrições |
| TrilhoId (PK) | Numérico | Indica o ID de um trilho | [0000, 9999] |  | Concebido pelo sistema / Não Alterável |
| Nome\_Trilho | String | Indica o nome de um trilho | Alfanumérico | Inserido pelo Professor de Desporto | Obrigatório/Alterável/ Preenchido pelo Professor de Desporto |
| Local\_Inicio\_Trilho | String | Indica o nome do inicio de um trilho | Alfanumérico | Inserido pelo Professor de Desporto | Obrigatório/Alterável/ Preenchido pelo Professor de Desporto |
| Local\_Fim\_Trilho | String | Indica o nome do fim de um trilho | Alfanumérico | Inserido pelo Professor de Desporto | Obrigatório/Alterável/ Preenchido pelo Professor de Desporto |
| Distancia\_Total | Numérico | Indica a distância total de um trilho | [000, 999] | Inserido pelo Professor de Desporto | Obrigatório/Alterável/ Preenchido pelo Professor de Desporto |
| Duracao\_Media | Numérico | Indica a duração média de um trilho | HH:MM:SS | Inserido pelo Professor de Desporto | Obrigatório/Alterável/ Preenchido pelo Professor de Desporto |
| Esta\_Ativo | Boolean | Indica se um determinado trilho está ativo ou desativo | Ativo, Desativo | Inserido pelo Professor de Desporto | Obrigatório/Alterável/ Preenchido pelo Professor de Desporto |
| Tempo\_Gasto | Numérico | Indica o tempo gasto a percorrer um trilho | HH:MM:SS | Inserido pelo Utilizador | Obrigatório/Alterável/ Preenchido pelo Utilizador |
| ProfessorId (FK) | Numérico | Indica o ID do professor | --------- | Botão ou opção equivalente | Não nulo / Alterável |
| DificuldadeId (FK) | String | Indica o ID da dificuldade | --------- | Botão ou opção equivalente | Não nulo / Alterável |

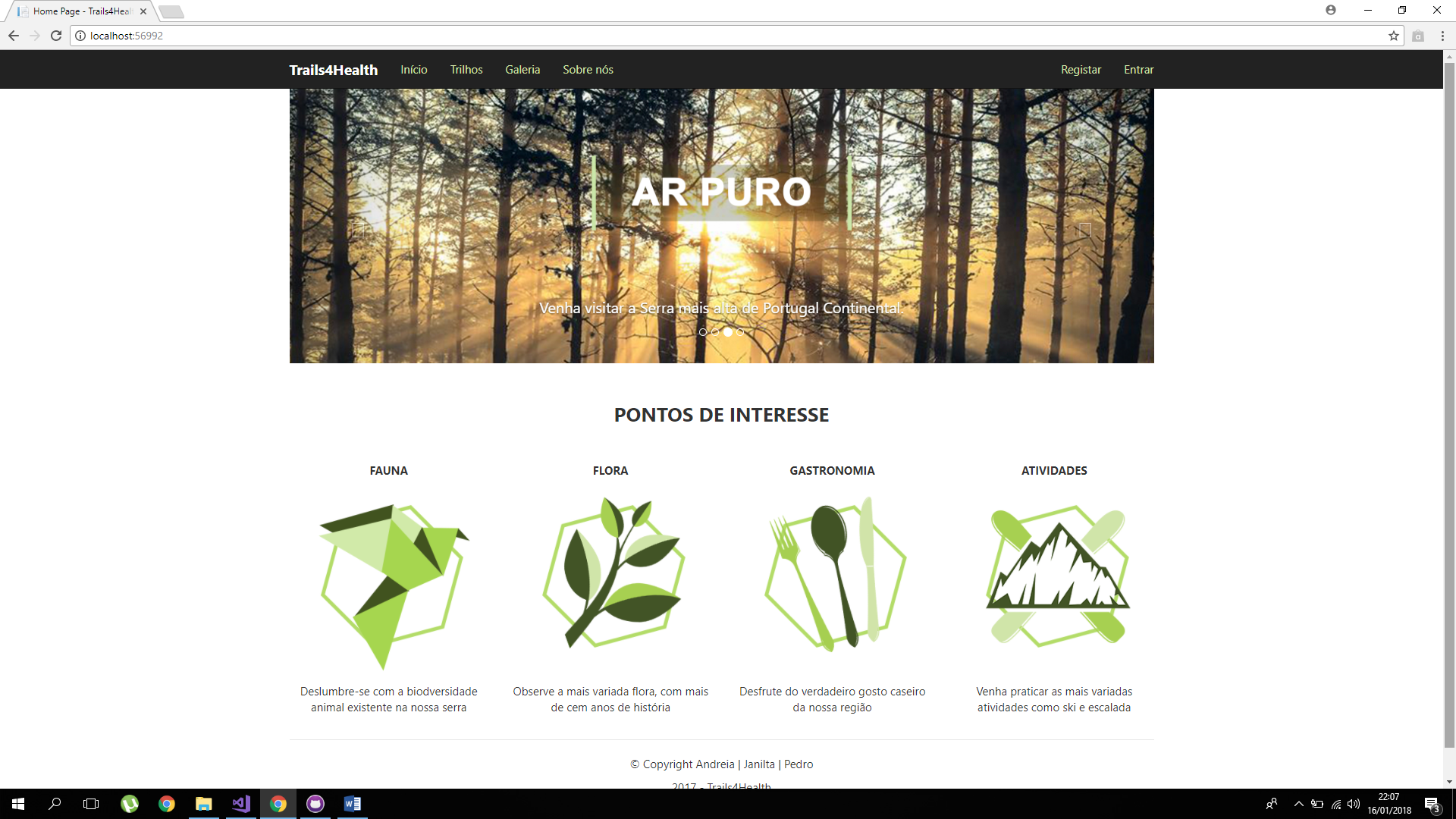
**Descrição das Operações:**

|  |  |
| --- | --- |
| **Operações** | |
| **Nome** | **Descrição** |
| **Consultar Trilho** | Operação que permite consultar trilhos existentes através do Nome\_Trilho. |

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

1. Consultar histórico de trilhos percorridos;
2. Configurar Trilho;
3. Consultar utilizadores que percorreram determinado trilho;
4. Agendar Trilho;
5. Alterar Agendamento do Trilho;
6. Consultar Trilho Agendado;
7. Cancelar Trilho Agendado;
8. Atualizar Estado Agendamento do Trilho.
9. Protótipo

# Página inicial

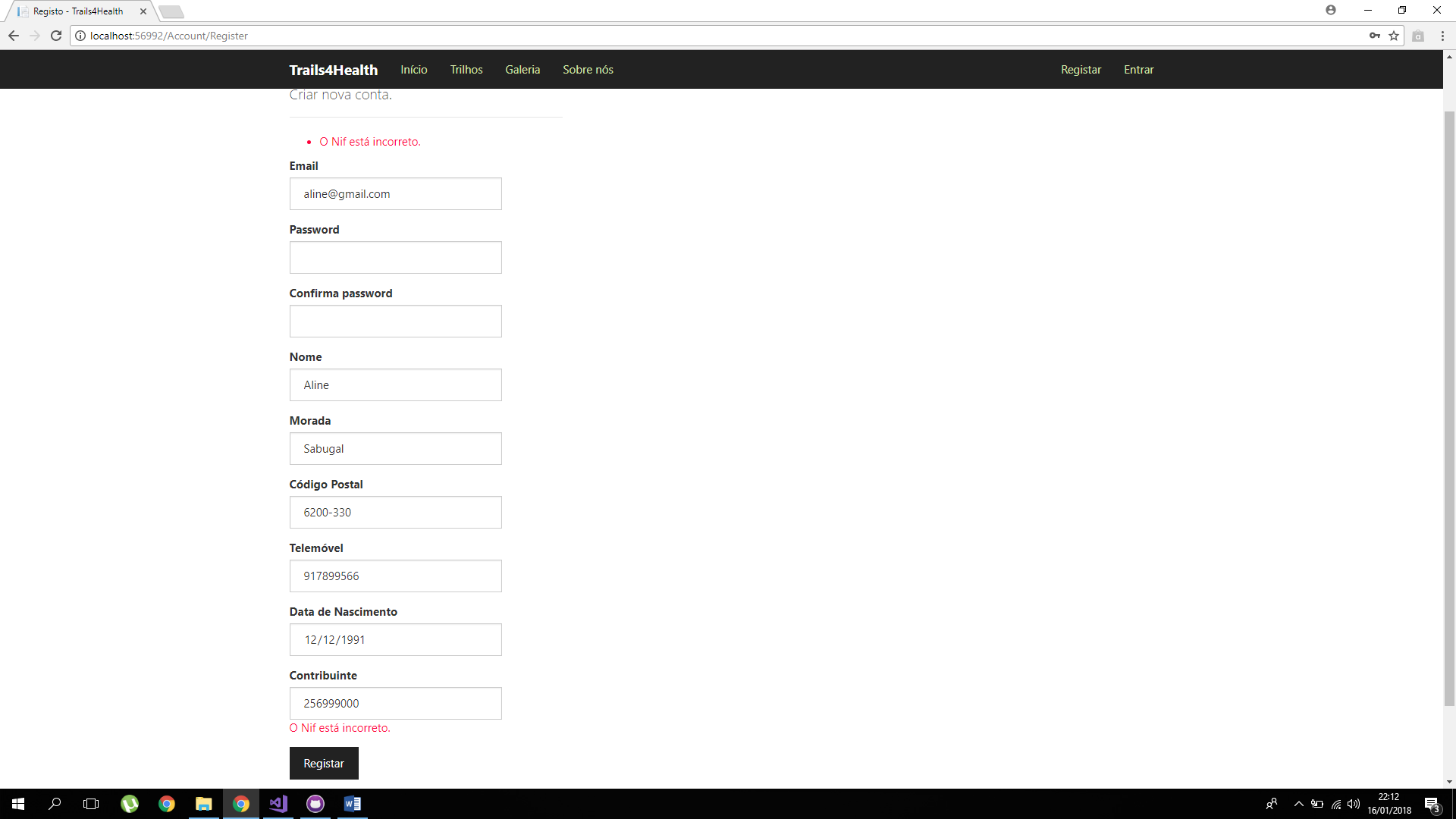


# Página de registo

Página com formulário de registo



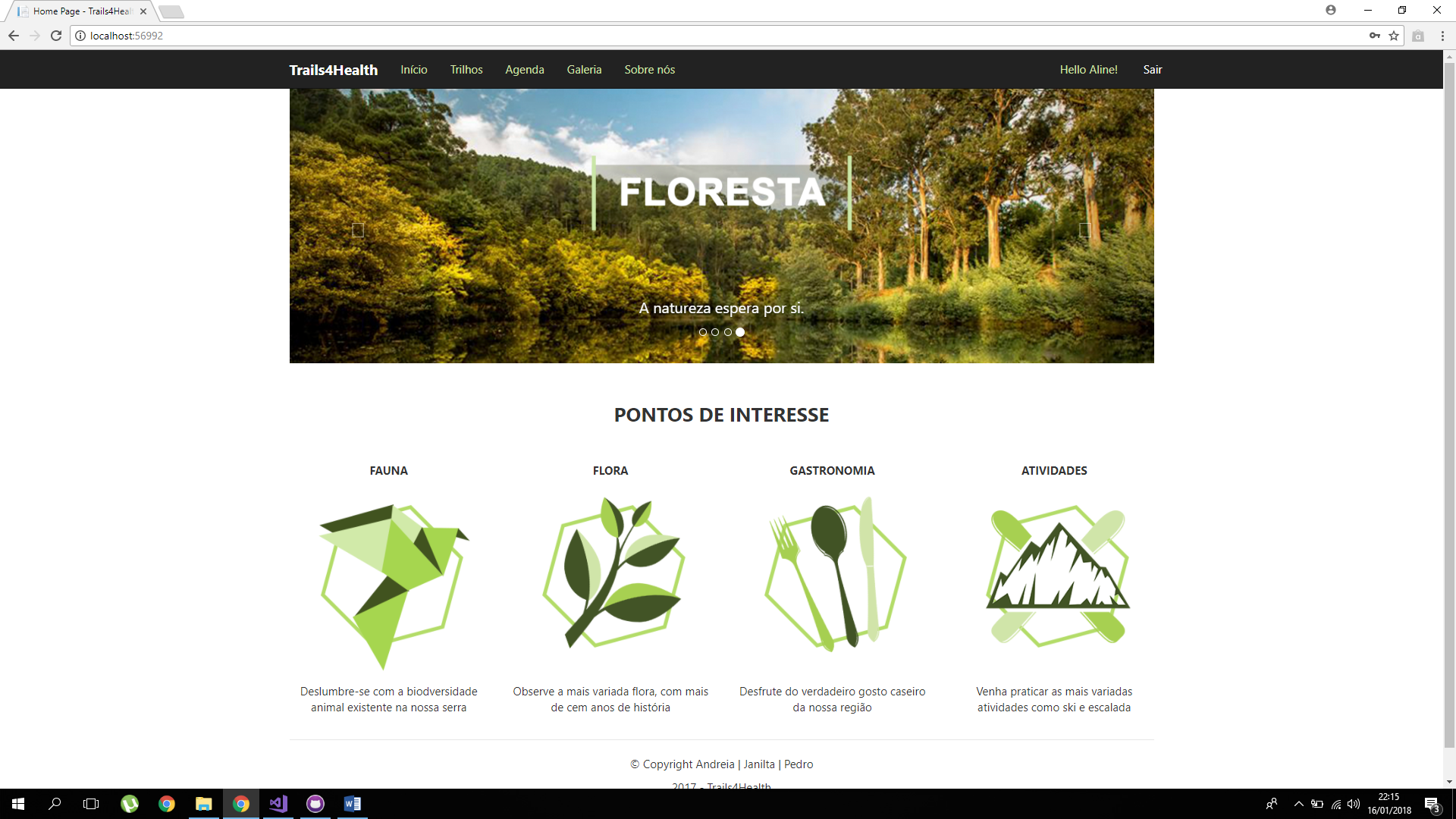
Página com NIF inválido através de digito de controlo



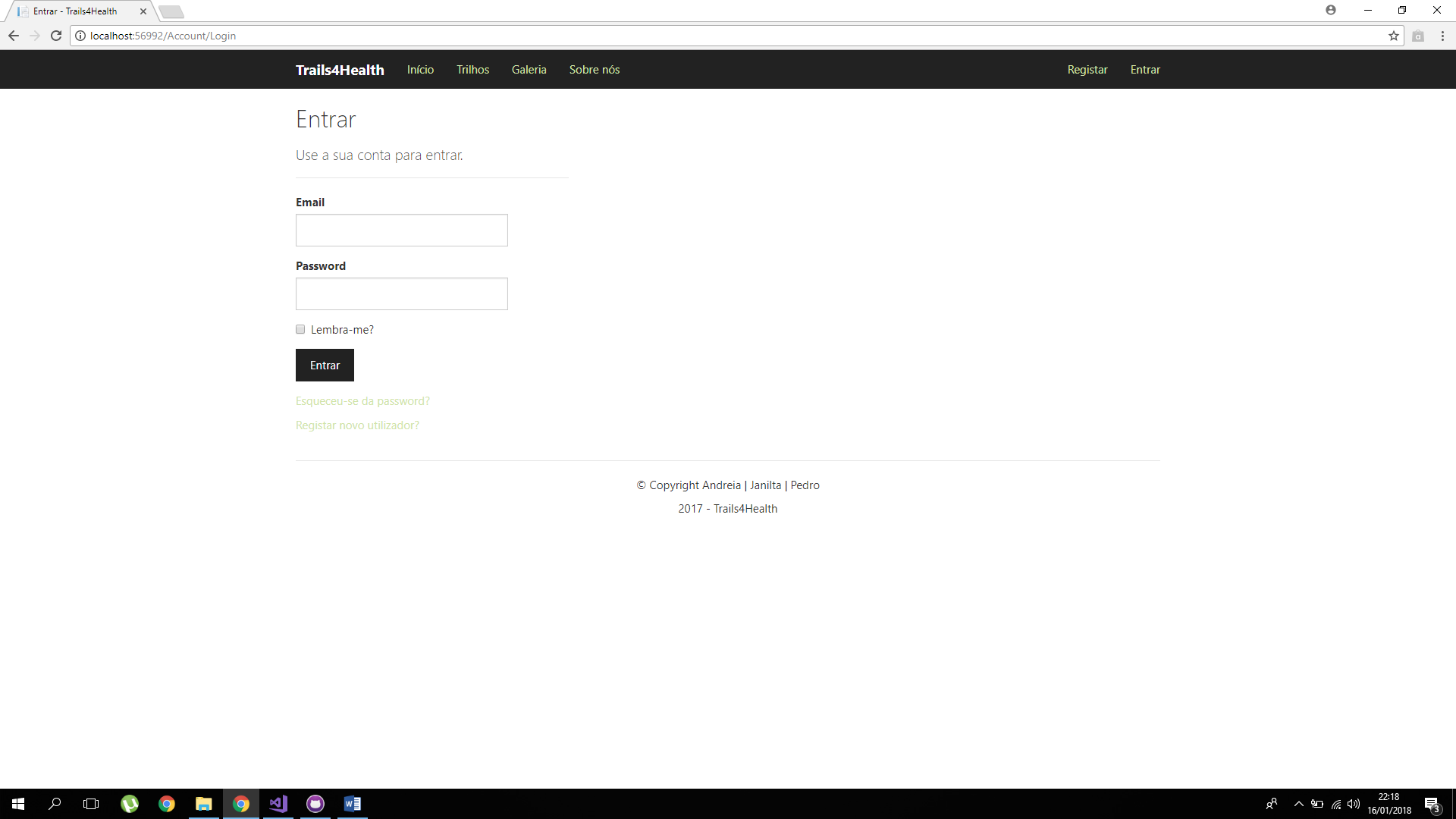
# Página após registo

Após registo é mostrada uma página que dá as boas vindas ao utilizador. A página mostra os menus a que o utilizador tem privilégios.

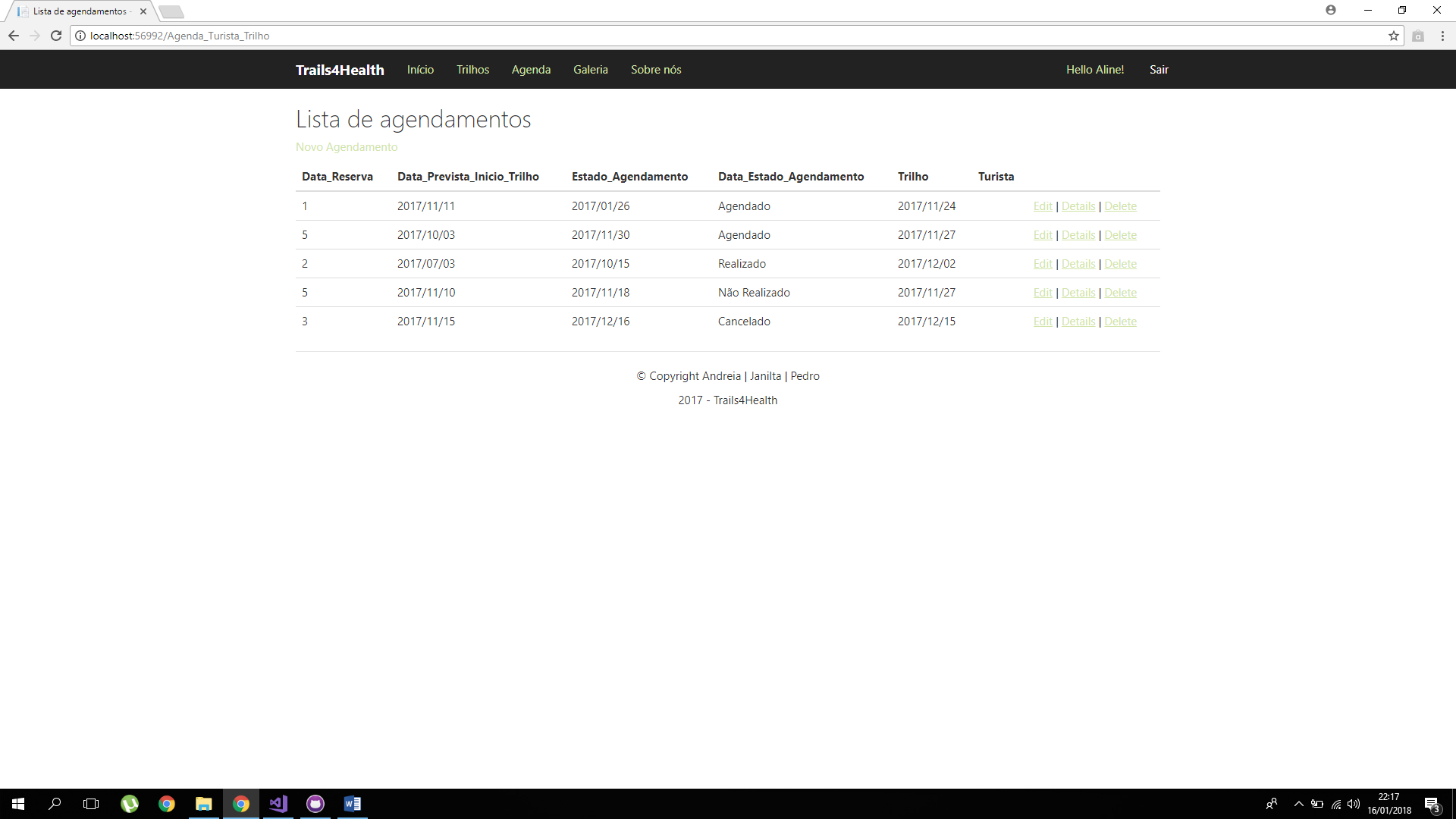
Por exemplo o menu de acesso à página de gestão de professores não é mostrado.



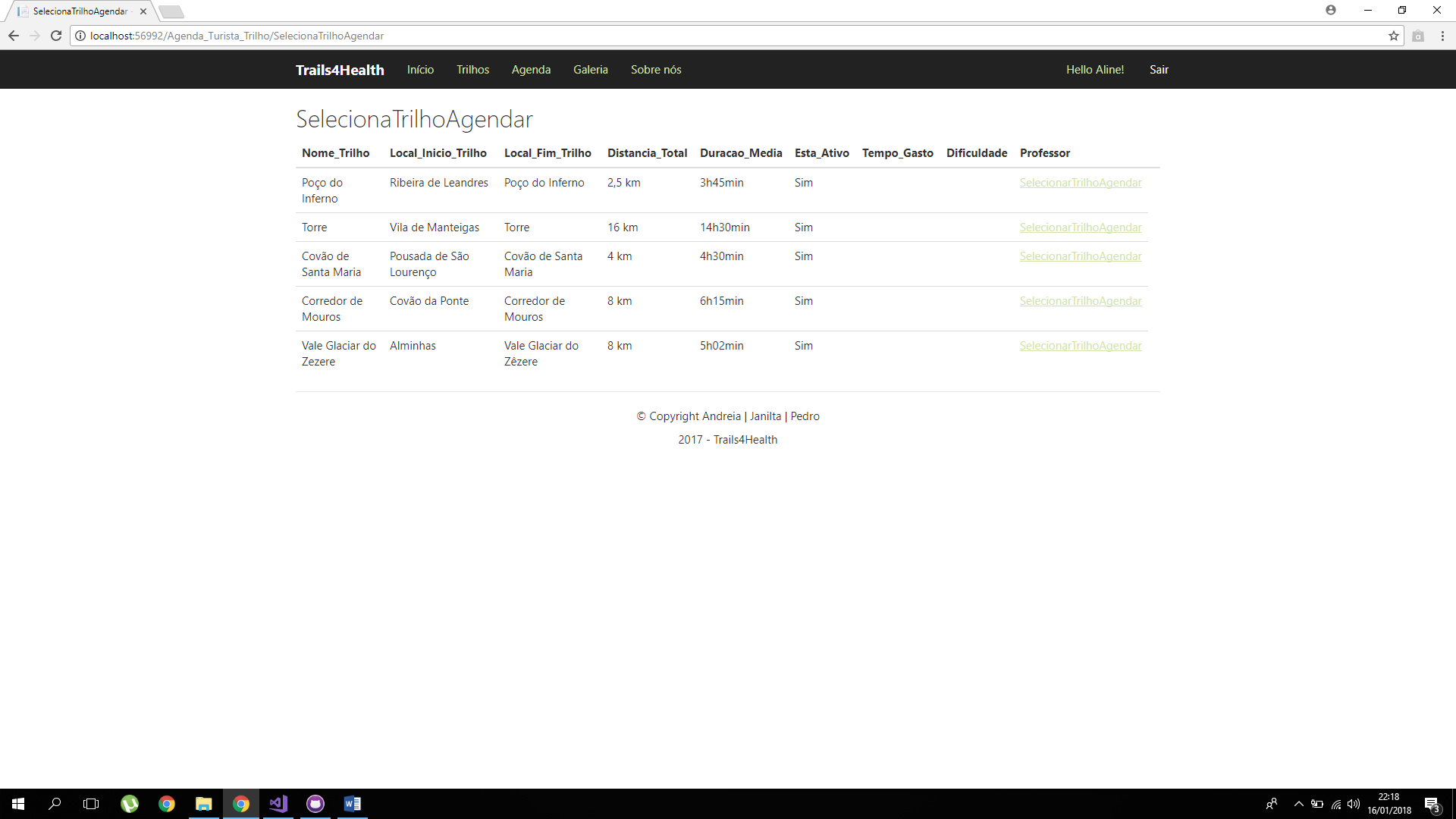
# Página Login



# Página Agendamentos



# Página Novo agendamento



Conclusão

O projeto desenvolvido revelou-se crucial, dinâmico e enriquecedor para consolidar a matéria dada na unidade curricular de Engenharia de Software II e Programação para a Internet. A divisão do processo de desenvolvimento em etapas facilita bastante a criação do projeto. Permite ao programador iniciar uma descrição abstrata do problema e que gradualmente migre para representações mais detalhadas do mesmo, até chegar ao código fonte. Posto isto, todo este processo que antecede a geração do código final é de extrema importância para o sistema. Os diagramas apresentados como o diagrama de sequência, diagrama de pacotes, diagrama de instalação, diagrama de classes entre outros, mostram-se capazes de atender as necessidades do sistema em questão, pelo fato de ilustrar de forma clara os objetos e ligações decorrentes de cada operação do sistema.

A modelagem do sistema foi uma fase bastante importante, principalmente no momento da implementação. Com os requisitos adquiridos e especificados através dos modelos da UML facilitou a visualização das funções oferecidas pelo sistema.

Posto isto, pensamos que o trabalho foi bem conseguido e que poderá ser usado num futuro próximo.

Anexos

# Atividades e tempos gastos

|  |  |  |  |
| --- | --- | --- | --- |
|  | Andreia Ernesto | Janilta Pires | Pedro Sanches |
| Lista de casos de uso | 2h | 2h | 2h |
| Tabela de comparação | 1h | 2h | 3h |
| Descrição de padrões de desenvolvimento | 5h | 4h | 2h |
| Descrição do tema | 30min | 30min | 1h |
| Diagrama de contexto | 30min | 30min | 30min |
| Diagrama de casos de uso | 3h | 3h | 3h |
| Descrição dos casos de uso | 10h | 8h | 10h |
| Diagramas de Sequência | 10h | 12h | 10h |
| Diagrama de Classes | 4h | 5h | 5h |
| Diagrama de Estados | 4h | 3h | 4h |
| Diagrama de Atividades | - | 2h | - |
| Diagrama de Componentes | 2h | - | - |
| Diagrama de Pacotes | 2h | - | 2h |
| Diagrama de Instalação | - | 2h | - |
| Semântica de classes | 4h | 4h | 4h |