

Instituto Politécnico da Guarda

Escola superior de Tecnologia e Gestão

# 

Relatório de Engenharia de Software II

### 

### 

### 

André Madeira

Daniel Carvalhinho

Ivo Pinto

# 

Índice

[Introdução: 1](#_Toc503910106)

[Descrição do tema do projeto 1](#_Toc503910107)

[Tabela Atividades e tempos gastos, em horas, por elemento de grupo 2](#_Toc503910108)

[Gráficos de Atividades e tempos gastos, em horas, por elemento de grupo 3](#_Toc503910109)

[Gráficos de Commits do GitHub 6](#_Toc503910110)

[Diagrama de Contexto 7](#_Toc503910111)

[Descrição dos Três Padrões 8](#_Toc503910112)

[Quitting Time 8](#_Toc503910113)

[Two Tier Review 9](#_Toc503910114)

[Spiral Development 9](#_Toc503910115)

[Estado de Arte 10](#_Toc503910116)

[Lista de casos de uso candidatos: 11](#_Toc503910117)

[Diagrama de Casos de Uso 12](#_Toc503910118)

[Diagrama de Casos de Uso Apenas com a Fronteira 13](#_Toc503910119)

[Casos de uso 13](#_Toc503910120)

[Tabela de Atores 14](#_Toc503910121)

[Descrição Casos de uso 14](#_Toc503910122)

[Editar Trilho 14](#_Toc503910123)

[Criar trilho 15](#_Toc503910124)

[Consultar trilho 15](#_Toc503910125)

[Criar etapa 16](#_Toc503910126)

[Classificar Imagem dos Trilhos 16](#_Toc503910127)

[Comparar Trilhos 17](#_Toc503910128)

[Desativar Trilho 17](#_Toc503910129)

[Diagrama Sequencia - Alterar Trilho 18](#_Toc503910130)

[Diagrama Sequencia - Criar Trilho 19](#_Toc503910131)

[Diagrama Sequencia - Comparar Trilho 20](#_Toc503910132)

[Diagrama Sequencia - Desativar Trilho 21](#_Toc503910133)

[Diagrama Sequencia - Consultar Trilho 22](#_Toc503910134)

[Diagrama de Classes 23](#_Toc503910135)

[Diagrama de Estados do Trilho 24](#_Toc503910136)

[Semântica das Classes 25](#_Toc503910137)

[Classe Estado 25](#_Toc503910138)

[Classe EtapasTrilhos 25](#_Toc503910139)

[Classe Fotos 26](#_Toc503910140)

[Classe TipoFoto 27](#_Toc503910141)

[Classe Dificuldade 27](#_Toc503910142)

[Classe EstadosTrilhos 28](#_Toc503910143)

[Classe Etapas 29](#_Toc503910144)

[Classe FotosTrilhos 30](#_Toc503910145)

[Classe Localizações 31](#_Toc503910146)

[Classe Trilho 32](#_Toc503910147)

[Diagrama de Atividades 34](#_Toc503910148)

[Diagrama de Componentes 35](#_Toc503910149)

[Diagrama de Pacotes de Casos de Uso 36](#_Toc503910150)

[Diagrama de Casos de uso do Pacote Professor 37](#_Toc503910151)

[Diagrama de Casos de uso do Pacote Turista 38](#_Toc503910152)

[Diagrama de Instalação 39](#_Toc503910153)

[Print screen das tabelas da Base de dados 40](#_Toc503910154)

[Print screen dos dados da Base de dados 43](#_Toc503910155)

[Protótipos da aplicação 46](#_Toc503910156)

[Conclusão 50](#_Toc503910157)

[Anexos 51](#_Toc503910158)

Índice Ilustrações

[Figura 1 Diagrama de Contexto 7](#_Toc503910159)

[Figura 2 Tabela Estado de Arte 10](#_Toc503910160)

[Figura 3 Diagrama de Casos de uso 12](#_Toc503910161)

[Figura 4 Diagrama de Casos de uso apenas com a Fronteira 13](#_Toc503910162)

[Figura 5 Diagrama de Sequencia Alterar Trilho 18](#_Toc503910163)

[Figura 6 Diagrama de Sequencia Criar Trilho 19](#_Toc503910164)

[Figura 7 Diagrama de Sequencia Comparar Trilho 20](#_Toc503910165)

[Figura 8 Diagrama de Sequencia Desativar Trilho 21](#_Toc503910166)

[Figura 9 Diagrama de Sequencia Desativar Trilho 22](#_Toc503910167)

[Figura 10 Diagrama de Classes 23](#_Toc503910168)

[Figura 11 Diagrama de Estados 24](#_Toc503910169)

[Figura 12 Diagrama de Atividades 34](#_Toc503910170)

[Figura 13 Diagrama de Componentes 35](#_Toc503910171)

[Figura 14 Diagrama de Pacotes de casos de uso 36](#_Toc503910172)

[Figura 15 Diagrama de Casos de Uso do Professor 37](#_Toc503910173)

[Figura 16 Diagrama de Casos de Uso do Turista 38](#_Toc503910174)

[Figura 17 Diagrama de Instalação 39](#_Toc503910175)

[Figura 18 Tabela Dificuldades (BaseDados) 40](#_Toc503910176)

[Figura 19 Tabela EstacoesAno (BaseDados) 40](#_Toc503910177)

[Figura 20 Tabela Estados (BaseDados) 40](#_Toc503910178)

[Figura 21 Tabela EstadosTrilhos (BaseDados) 40](#_Toc503910179)

[Figura 22 Tabela Etapas (BaseDados) 40](#_Toc503910180)

[Figura 23 Tabela EtapasTrilhos (BaseDados) 41](#_Toc503910181)

[Figura 24 Tabela Fotos (BaseDados) 41](#_Toc503910182)

[Figura 25 Tabela Trilhos (BaseDados) 41](#_Toc503910183)

[Figura 26 Tabela Localizacao (BaseDados) 41](#_Toc503910184)

[Figura 27 Tabela TiposFotos (BaseDados) 42](#_Toc503910185)

[Figura 28 Tabela Trilhos (BaseDados) 42](#_Toc503910186)

[Figura 29 Tabela Dificuldades (BaseDados Preenchida) 43](#_Toc503910187)

[Figura 30Tabela EstacoesAno (BaseDados Preenchida) 43](#_Toc503910188)

[Figura 31Tabela Estados (BaseDados Preenchida) 43](#_Toc503910189)

[Figura 32Tabela EstadosTrilhos (BaseDados Preenchida) 43](#_Toc503910190)

[Figura 33 Tabela Etapas (BaseDados Preenchida) 44](#_Toc503910191)

[Figura 34 Tabela EtapasTrilhos (BaseDados Preenchida) 44](#_Toc503910192)

[Figura 35 Tabela Fotos (BaseDados Preenchida) 44](#_Toc503910193)

[Figura 36 Tabela FotosTrilhos (BaseDados Preenchida) 44](#_Toc503910194)

[Figura 37 Tabela Localizacoes (BaseDados Preenchida) 45](#_Toc503910195)

[Figura 38 Tabela TiposFotos (BaseDados Preenchida) 45](#_Toc503910196)

[Figura 39 Tabela Trilhos (BaseDados Preenchida) 45](#_Toc503910197)

[Figura 40 Index (Pagina Inicial) 46](#_Toc503910198)

[Figura 41Eliminar Fotos 46](#_Toc503910199)

[Figura 42 Detalhes Foto 47](#_Toc503910200)

[Figura 43 Editar Foto 47](#_Toc503910201)

[Figura 44 Index Fotos 47](#_Toc503910202)

[Figura 45 Index Trilho 48](#_Toc503910203)

[Figura 46 Criar Trilho 48](#_Toc503910204)

[Figura 47 Consultar Trilho 48](#_Toc503910205)

[Figura 48 Editar Trilho 49](#_Toc503910206)

[Figura 49 Comparar Trilho 49](#_Toc503910207)

# Introdução:

No âmbito das cadeiras de Engenharia de Software e de Programação para a Internet, foi-nos proposta a realização de um projeto, em UML, bem como um protótipo, em ASP.NET MVC, recorrendo à técnica de desenvolvimento ágil. O seguinte relatório vai descrever todas as funcionalidades do produto desenvolvido durante o semestre.

## Descrição do tema do projeto

Este projeto nasce do desafio lançado pela Associação Geoparque Estrela e pelo Parque Natural da Serra da Estrela, que objetiva caracterizar o esforço de pedestrianismo, ao longo de uma grande rota, que no futuro seja sustentada por indicadores técnicos, físicos e digitais.” De maneira a sustentar digitalmente este projeto nasce a aplicação Trails4Health.

A aplicação deve permitir a criação, edição, eliminação e comparação de trilhos por parte da administração (professores de educação física), e que estes sejam divididos em etapas devido à larga extensão que se espera cobrir. Serão avaliados indicadores fisiológicos (esforço cardíaco e gasto energético) e biomecânicos (impacto articular e muscular) discriminadores do esforço requerido para completar cada etapa, de maneira a possibilitar a aplicação identificar e sugerir aos seus utilizadores as etapas e trilhos que estão de acordo com a sua idade e/ou nível de aptidão física avaliados pelo preenchimento de um questionário desenvolvido por especialistas da área.

De maneira a sustentar a escolha do utilizador, a aplicação deverá permitir a visualização de várias características dos trilhos disponíveis. Será possível aos utilizadores visualizar fotos, marcos, pontos de interesse como cafés e hotéis, entre outros com recurso a geolocalização.

Os utilizadores registados vão conseguir visualizar os trilhos já percorridos, dar opinião e classificá-los, permitindo identificar os trilhos mais populares e os menos utilizados, que poderão ser alterados pela administração para melhorar a adesão aos mesmos.

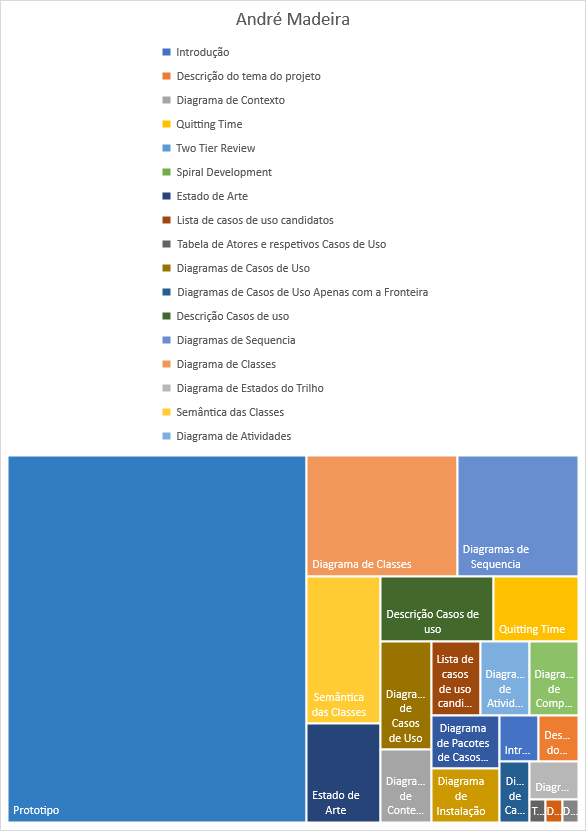
Este projeto foi dividido em módulos, sendo que, este relatório tem como objetivo principal descrever os seguintes pontos da aplicação:

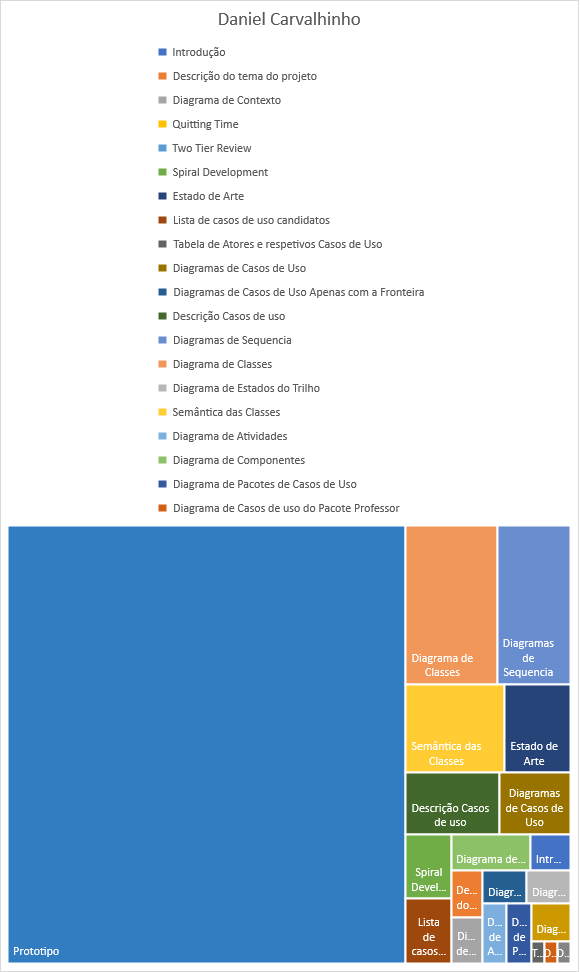
* Criação, Edição e Eliminação de Trilhos
* Comparação entre Trilhos
* Classificação de Imagens

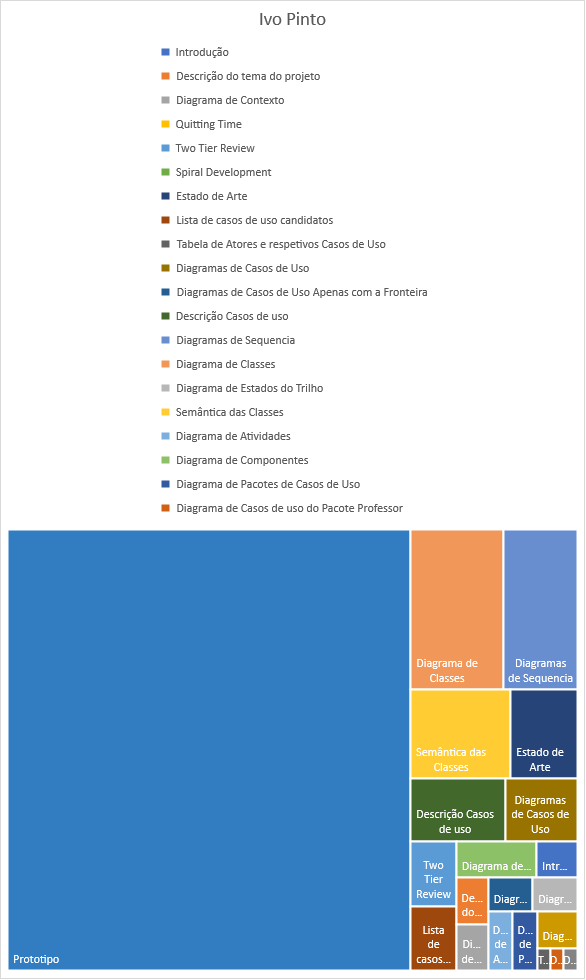
|  |  |  |  |
| --- | --- | --- | --- |
| **Tarefa** | Nº Horas Gastas | | |
| André Madeira | Daniel Carvalhinho | Ivo Pinto |
| Introdução | 1 | 1 | 1 |
| Descrição do tema do projeto | 1 | 1 | 1 |
| Diagrama de Contexto | 2 | 1 | 1 |
| Quitting Time | 3 | 0 | 0 |
| Two Tier Review | 0 | 0 | 2 |
| Spiral Development | 0 | 2 | 0 |
| Estado de Arte | 4 | 4 | 4 |
| Lista de casos de uso candidatos | 2 | 2 | 2 |
| Tabela de Atores e respetivos Casos de Uso | 0,2 | 0,2 | 0,2 |
| Diagramas de Casos de Uso | 3 | 3 | 3 |
| Diagramas de Casos de Uso Apenas com a Fronteira | 1 | 1 | 1 |
| Descrição Casos de uso | 4 | 4 | 4 |
| Diagramas de Sequencia | 8 | 8 | 8 |
| Diagrama de Classes | 10 | 10 | 10 |
| Diagrama de Estados do Trilho | 1 | 1 | 1 |
| Semântica das Classes | 6 | 6 | 6 |
| Diagrama de Atividades | 2 | 1 | 1 |
| Diagrama de Componentes | 2 | 2 | 2 |
| Diagrama de Pacotes de Casos de Uso | 2 | 1 | 1 |
| Diagrama de Casos de uso do Pacote Professor | 0,2 | 0,2 | 0,2 |
| Diagrama de Casos de uso do Pacote Turista | 0,2 | 0,2 | 0,2 |
| Diagrama de Instalação | 2 | 1 | 1 |
| Protótipo | 60 | 120 | 120 |
| Total: | 114,6 | 169,6 | 169,6 |

# Tabela Atividades e tempos gastos, em horas, por elemento de grupo

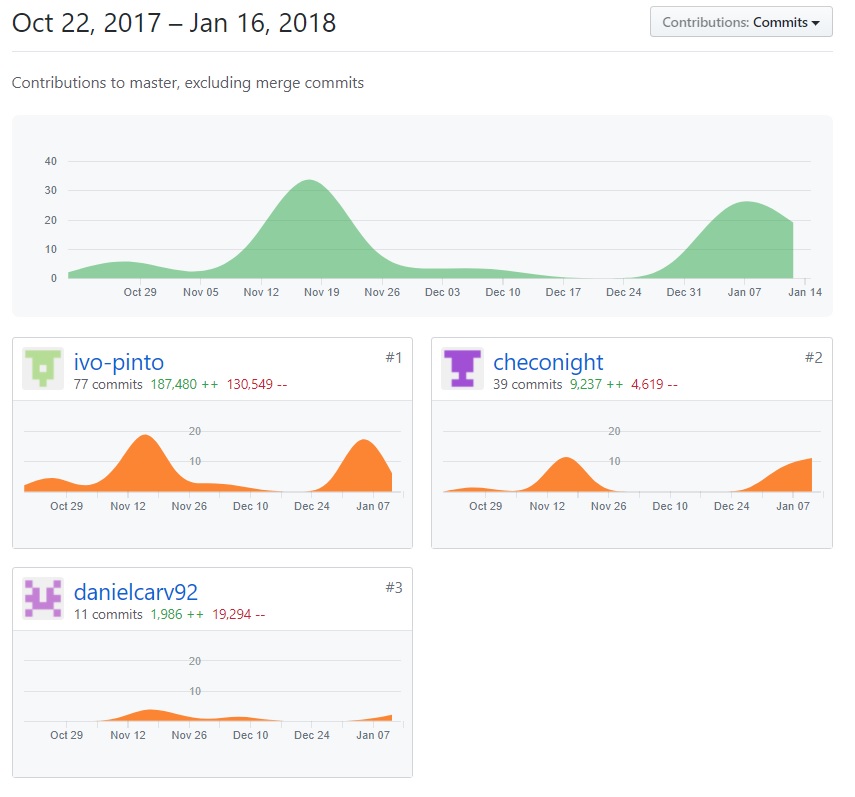
# Gráficos de Atividades e tempos gastos, em horas, por elemento de grupo







# Gráficos de Commits do GitHub



Legenda:

Ivo-pinto – Ivo Pinto

Danielcarv92 – Daniel Carvalhinho

Checonight – André Madeira

# Diagrama de Contexto

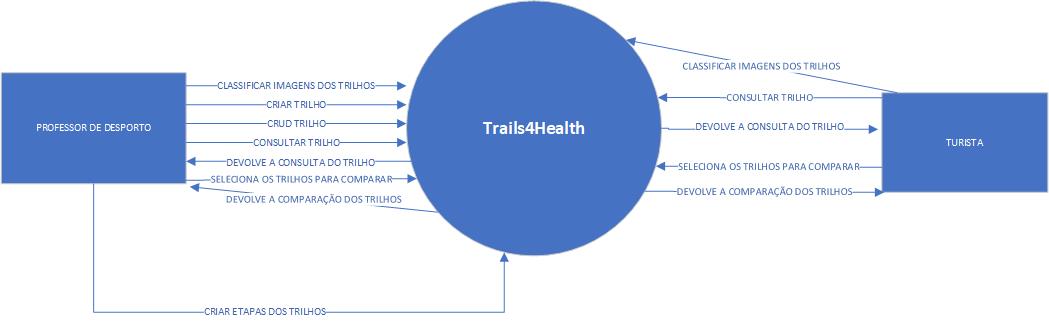


Figura 1 Diagrama de Contexto

# Descrição dos Três Padrões

## Quitting Time

Desenvolver casos de uso desnecessários é um desperdício de tempo, recursos e pode ainda atrasar o projeto. Os requisitos devem ser especializados para reduzir o risco de falhanço do projeto e devemos seguir esses procedimentos rigorosamente para correr tudo como o cliente quer.

Os casos de uso para serem bons tem de ser equilibrados e sobretudo, devem ter o essencial para o projeto funcionar. Por vezes é complicado deixar de escrever casos de uso porque temos de alternar e não dificultar o processo porque por vezes temos de escolher entre uns casos e outros para não acrescentar demasiada informação, muita dela inútil. Devemos evitar que um cliente queira uma alteração após os casos de uso estarem bem definidos, dai a necessidade de alguma experiencia e delinear bem o projeto para evitar ter que voltar atrás no projeto e perder tempo.

A experiencia ajuda a que não aconteçam este tipo de erros. Os requisitos não devem ser específicos de mais, pois os mesmos podem comprometer os pré-requisitos. A deteção inicial de um erro num projeto é muito melhor pois evitamos ter que recomeçar o projeto e perder tempo e dinheiro com isso, pois é mau para o cliente e para o programador. Não devemos adicionar casos de uso a mais do que aqueles que já foram aprovados pelo cliente. Estas são as três perguntas para que um cliente faz para saber se o projeto cumpre os requisitos ou não:

1. Foram identificados e documentados todos os objetivos e todos os atores?
2. O cliente ou o seu representante reconheceu e aprovou todos os casos de uso antes o desenvolvimento do projeto?
3. O designer pode implementar os casos de uso?

A resposta a estas perguntas deve ser sim para avançar com o projeto, caso contrario ainda existe trabalho a fazer.

Exemplo:   
Wings Over The World (Asas sobre o mundo). Os casos de uso não estão completos ate estar tudo bem detalhado e discutido. Não se deve perder tempo e avançar com o projeto sem quaisquer preocupações.

“Quando é que se sabe quando os casos de uso estão completos?”   
O QuittingTime equilibra o risco do atraso do projeto com o risco da falta de requerimentos.  
É necessário ter sempre em conta sempre cuidado o que se faz e implementa. A comunicação entre o cliente e o programador deve ser bastante clara e esclarecedora para que não haja erros e para que não haja falsas interpretações, para o produto final corresponder aos requisitos do cliente.

## Two Tier Review

O Two Tier Review é um método que tem como principais objetivos: O valor comercial, a validade das especificações, a implementação adequada aos meios e o rigor de conteúdo. é um método de revisão de casos de uso, dividido por duas camadas. Este método apesar de contemplar revisões desde o início do projeto, como estas têm um custo elevado é aconselhado que se façam em na menor quantidade possível com o menor número de pessoas possível, e por pessoas que não o próprio autor, num grupo participativo.

Quando este método é aplicado a sistemas muito grandes e complexos, deve-se fazer revisões realizadas por grupos de várias funções. Após estas revisões terminarem, um grupo externo deve proceder a uma avaliação geral em que engloba todos os relatórios dos diferentes grupos internos. Desta maneira a camada interna foca-se em avaliar o funcionamento do sistema de maneira a que a camada externa consiga juntar todas as funcionalidades, abstraindo-se dos problemas mais específicos e pensando no global.

## Spiral Development

Desenvolver casos de uso de uma só vez, é uma tarefa complicada e cara, que torna a adição de nova informação nos mesmos. O que pode adiar a deteção de fatores de risco.

O padrão de desenvolvimento Spiral Development apresenta a resolução deste problema usando um método iterativo.

Escrever os casos de uso através de iterações permite facilmente a correção de erros ou alguma alteração necessária, caso a informação atual esteja incorreta, perdendo muito menos tempo e trabalho do que outros métodos em que os que desenvolvem todos os casos de uso de uma só vez. No entanto, neste padrão, tem que haver um limite de tempo para a edição dos casos de uso. Quando se têm todos os casos de uso para satisfazer todos os requisitos, deve-se parar a edição dos mesmos.

A recolha de requisitos é vital para o sucesso de um projeto, apesar de ser apenas uma parte do mesmo, pelo que estes devem estar prontos assim que possível porque são necessários para outras partes do projeto.

Á medida que estudamos os casos de uso é provável que estes sofram alterações.

Os erros das funcionalidades podem tornar-se dispendiosos, dependendo da fase em que são encontrados, se for no inicio do projeto estes não têm grande impacto. Mas se for numa fase mais avançada estes poderão ter custos significativos.

# Estado de Arte

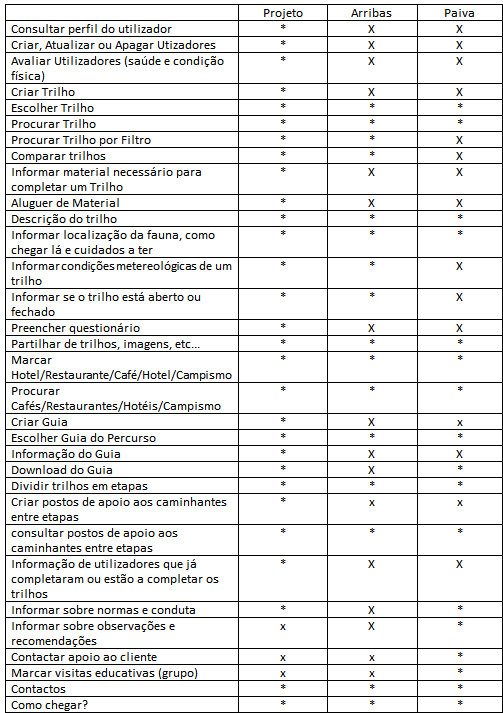


Figura 2 Tabela Estado de Arte

# 

# Lista de casos de uso candidatos:

* Consultar perfil do utilizador
* Criar, Atualizar ou Apagar Utizadores
* Avaliar Utilizadores (saúde e condição física)
* Criar Trilho
* Escolher Trilho
* Procurar Trilho
* Procurar Trilho por Filtro:
* Dificuldade;
* Duração Média;
* Inclinação;
* Distancia;
* Forma; (circular)
* Elevação;
* Preço;
* Material.
* Comparar trilhos;
* Informar material necessário para completar um Trilho (material de escalada, canoagem, sky, snowboard, bicicleta, pesca, parapente, campismo.)
* Aluguer de Material
* Descrição do trilho (Imagens, texto, Fauna, Flora, Historial,Geologia,Arqueologia)
* Informar localização da fauna, como chegar lá e cuidados a ter
* Informar condições metereológicas de um trilho
* Informar se o trilho está aberto ou fechado (consoante condições metereológicas por exemplo)
* Preencher questionário (Avaliação do trilho por parte dos utilizadores)
* Partilhar de trilhos, imagens, etc...
* Marcar Hotel/Restaurante/Café/Hotel/Campismo/Etc entre trilhos
* Procurar Cafés/Restaurantes/Hotéis/Campismo/Etc entre trilhos
* Criar Guia
* Escolher Guia do Percurso
* Informação do Guia
* Download do Guia
* Dividir trilhos em etapas
* Criar postos de apoio aos caminhantes entre etapas
* Informar sobre normas e conduta
* Informação de utilizadores que já completaram ou estão a completar os trilhos
* Contactos
* Como chegar?

# Diagrama de Casos de Uso

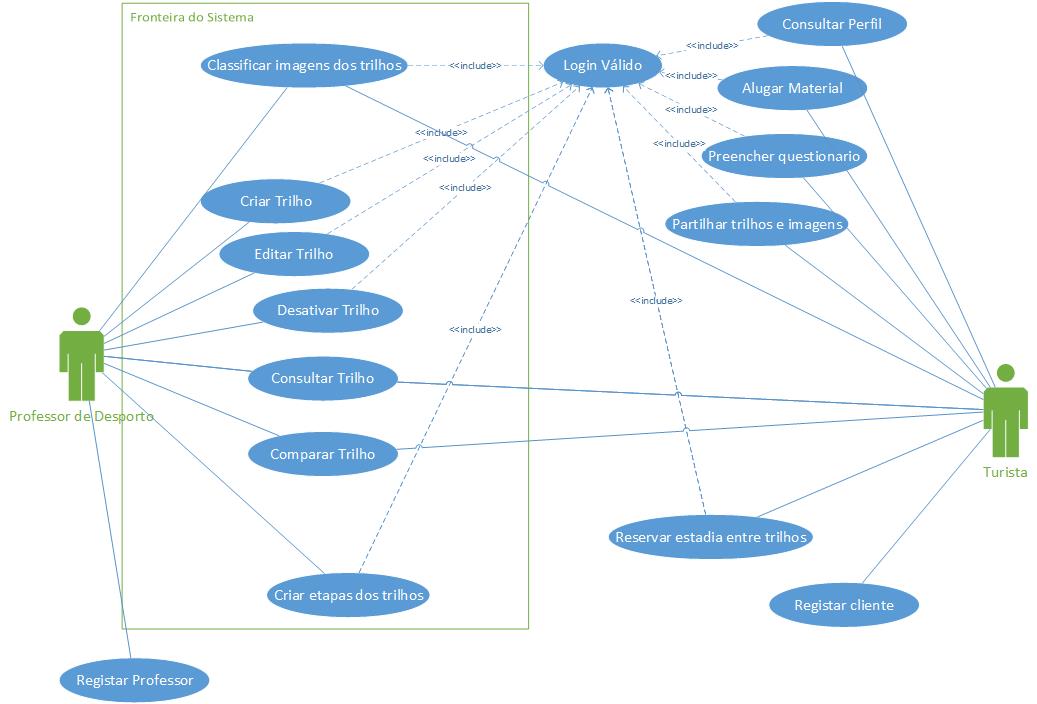


Figura 3 Diagrama de Casos de uso

# Diagrama de Casos de Uso Apenas com a Fronteira

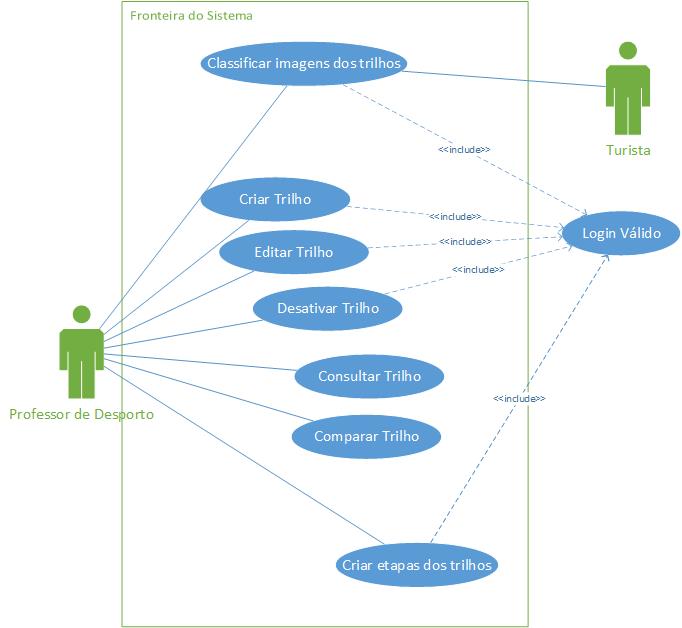


Figura 4 Diagrama de Casos de uso apenas com a Fronteira

# 

# Casos de uso

* Criar Trilho
* Alterar Trilho
* Desativar Trilho
* Consultar Trilho
* Comparar Trilho
* Criar Etapa
* Classificar Imagem

# Tabela de Atores

|  |  |
| --- | --- |
| **Atores** | **O que faz?** |
| Professor | Classificar imagens dos Trilhos |
| Professor | Criar Trilho |
| Professor | Editar Trilho |
| Professor | Desativar Trilho |
| Professor | Consultar Trilho |
| Professor | Comparar Trilho |
| Professor | Criar Etapas dos Trilhos |
| Turista | Consultar Trilho |
| Turista | Comparar Trilho |

# Descrição Casos de uso

**Nome:**

## Editar Trilho

Pré Condição: O utilizador tem que ter privilégios de edição para editar o trilho.

**Caminho principal:**

1. O Professor realiza um pedido ao sistema para aceder á zona ‘Editar Trilho’.
2. O sistema devolve uma lista dos trilhos existentes.
3. O professor seleciona o trilho que deseja editar, clicando no botão ‘Editar Trilho’.
4. O sistema devolve o formulário com os dados do trilho a alterar.
5. O professor altera os dados desejados e clica no botão alterar.
6. O sistema devolve uma mensagem de confirmação da alteração do trilho.

**Caminhos alternativos:**

5.a) O sistema devolve uma mensagem ao professor para preencher os campos obrigatórios.

**Suplementos ou adornos:**

Verificar se o professor preencheu todos os campos do formulário.

**Nome:**

## Criar trilho

**Descrição**   
 Permite ao professor de educação física criar um trilho

**Pré-condição**

O professor de desporto tem que estar logado   
 Tem que haver no mínimo uma etapa previamente criada  
**Caminho Principal**

1. O Professor realiza um pedido ao sistema para aceder a zona de criar trilho
2. O sistema devolve o formulário de criar trilhos ao professor
3. O professor preenche o formulário com os atributos do trilho e clica no botão "criar trilho"
4. O sistema devolve uma mensagem de confirmação da criação do trilho.

**Caminhos Alternativos:**

3. a) o sistema devolve uma mensagem ao professor para preencher os campos obrigatórios

**Suplementos e Adornos:**   
   
 verificar se o professor preencheu todos os campos obrigatórios do formulário   
 a descrição não pode exceder 1000 caracteres   
 calcular a dificuldade consoante a media de dificuldade das etapas

Pós-condição

## 

**Nome:**

## Consultar trilho

**Pré Condição:**

Não Tem.

**Caminho principal:**

1. O utilizador seleciona a opção ‘Trilho’ na HomePage
2. O sistema devolve um painel, com a lista dos trilhos marcados como visíveis, e com os seus atributos
3. O utilizador seleciona o Trilho a que pretende aceder
4. O sistema devolve uma página com as informações do trilho

**Caminhos alternativos:**

3.a) O utilizador pode regressar à Home Page

**Suplementos ou adornos:**

Verificar se o professor preencheu todos os campos do formulário.

# 

**Nome:**

## Criar etapa

**Pré Condição:**

O professor tem que ter privilégios de edição para editar o trilho.

**Caminho principal:**

1. O Professor realiza um pedido ao sistema para aceder à zona ‘Criar Etapa’
2. O sistema devolve o formulário de ‘Criar Etapa’ ao professor
3. O professor preenche o formulário com os atributos da etapa e clica no botão ‘Criar Etapa’

**Caminhos alternativos:**

3.a) O sistema devolve uma mensagem ao professor para preencher os campos obrigatórios

**Suplementos ou adornos:**

verificar se o professor preencheu todos os campos obrigatórios do formulário   
 a descrição não pode exceder 1000 caracteres

**Nome:**

## Classificar Imagem dos Trilhos

**Pré Condição:**

O utilizador tem que estar registado e logado no sistema

**Caminho principal:**

1. O utilizador faz um pedido ao sistema para aceder à zona de “Classificar Imagem”.
2. O sistema devolve um formulário com os atributos da imagem a preencher e uma opção para fazer upload da imagem.
3. O utilizador faz upload da imagem, preenche os seus atributos e carrega no botão “Confirmar”
4. O sistema devolve uma mensagem de sucesso ao utilizador.

**Caminhos alternativos:**

3.a) O utilizador pode cancelar a Acão

**Suplementos ou adornos:**

Verificar se o utilizador preenche todos os campos do formulário.

A imagem tem que ser validada pelo administrador do sistema.

**Nome:**

## Comparar Trilhos

**Pré Condição:**

Não Tem.

**Caminho principal:**

1. O utilizador faz um pedido para aceder à zona “Comparar Trilhos”
2. O sistema devolve um formulário com os trilhos visíveis na base de dados
3. O utilizador seleciona dois trilhos, e carrega no botão comprar
4. O sistema devolve um painel com os resultados da comparação

Caminhos alternativos:

3.a) O utilizador pode cancelar a Acão carregando no botão “cancelar”

**Suplementos ou adornos:**

Verificar se o utilizador selecionou dois trilhos.

**Nome:**

## Desativar Trilho

**Pré Condição:**

O utilizador tem que ter privilégios de edição para desativar o trilho.

**Caminho principal:**

1. O Professor realiza um pedido ao sistema para aceder à zona ‘Desativar Trilho’.
2. O sistema devolve uma lista dos trilhos existentes.
3. O professor seleciona o trilho que deseja desativar, clicando no botão ‘Desativar Trilho’.
4. O sistema devolve uma mensagem para confirmar a desativação do trilho.
5. O utilizador confirma a desativação do trilho.
6. O sistema devolve uma mensagem de sucesso.

**Caminhos alternativos:**

3.a) O professor pode cancelar a Acão.

5.a) O professor pode cancelar a Acão.

**Suplementos ou adornos:**

Um trilho tem que estar selecionado para que possa ser desativado.

# Diagrama Sequencia - Alterar Trilho

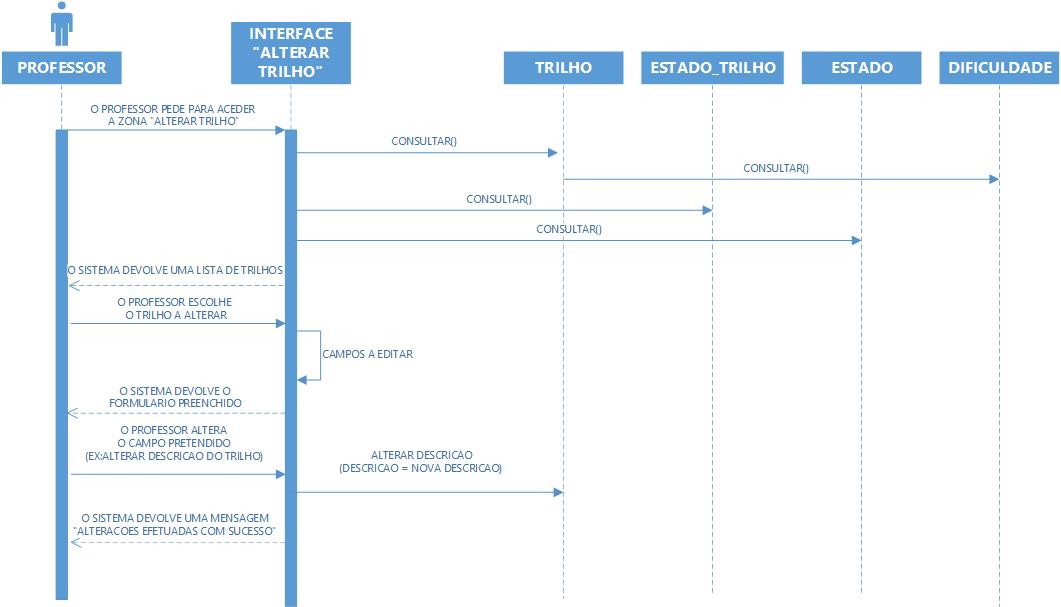


Figura 5 Diagrama de Sequencia Alterar Trilho

# Diagrama Sequencia - Criar Trilho

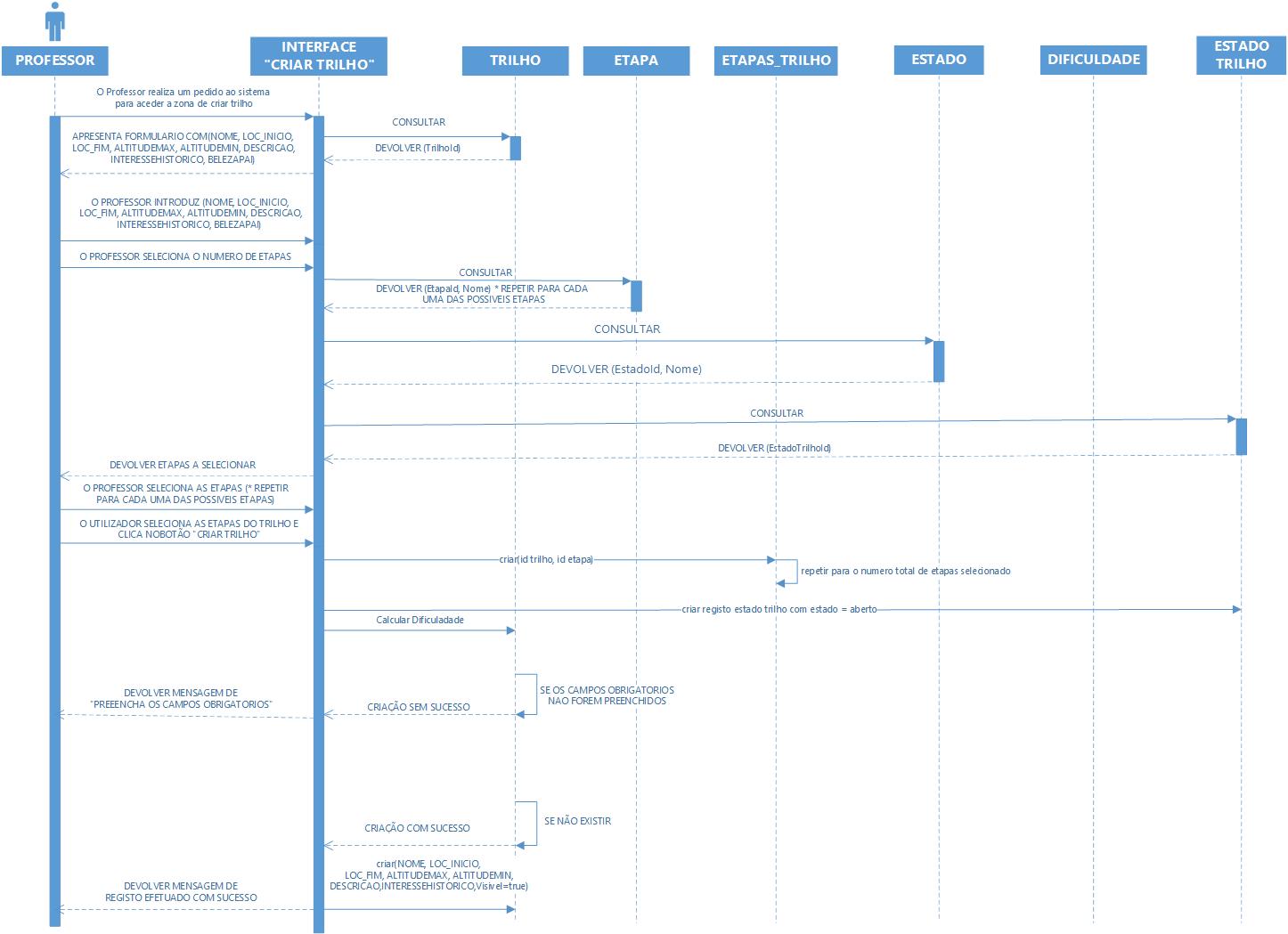


Figura 6 Diagrama de Sequencia Criar Trilho

# Diagrama Sequencia - Comparar Trilho

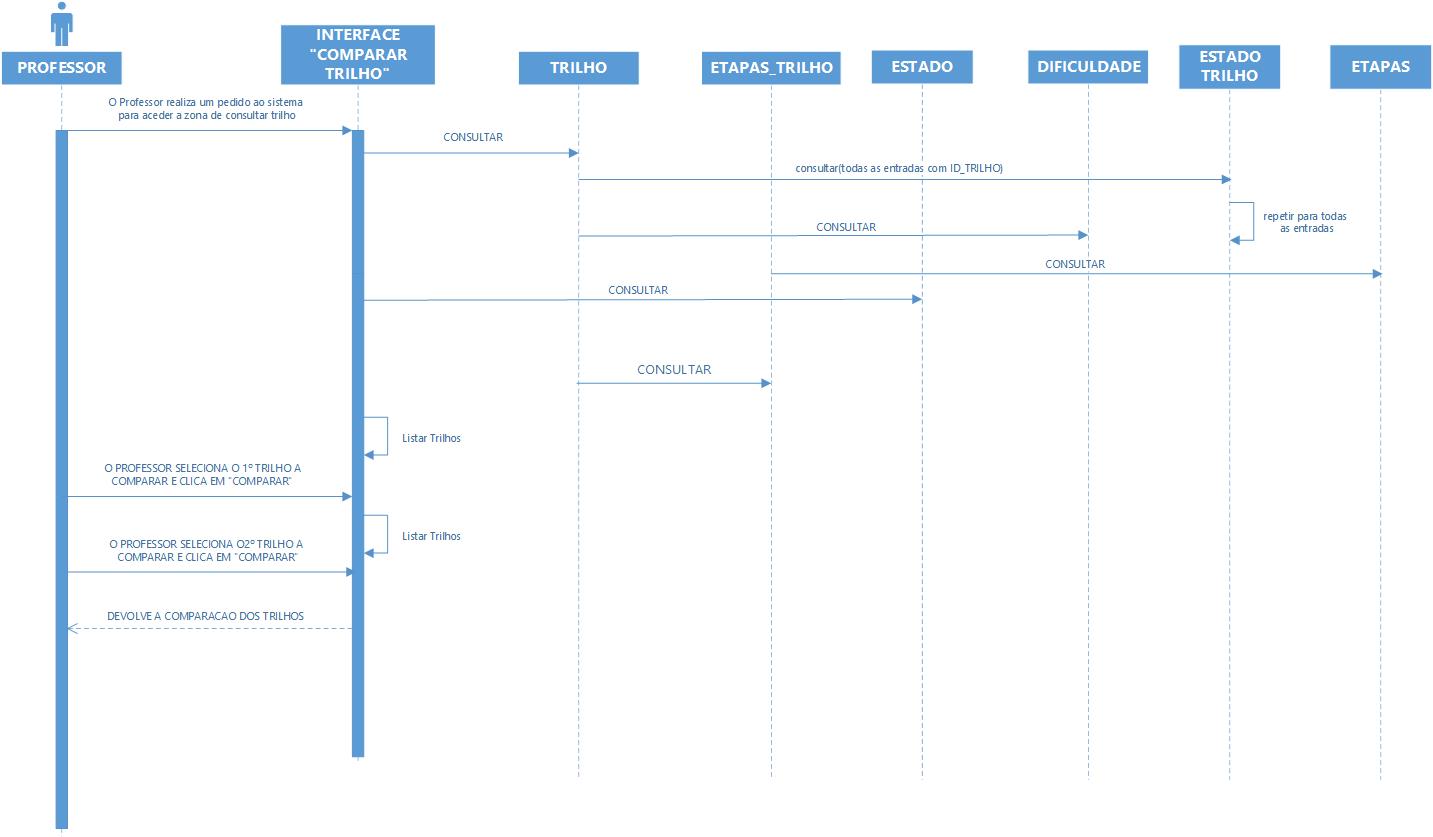


Figura 7 Diagrama de Sequencia Comparar Trilho

# Diagrama Sequencia - Desativar Trilho

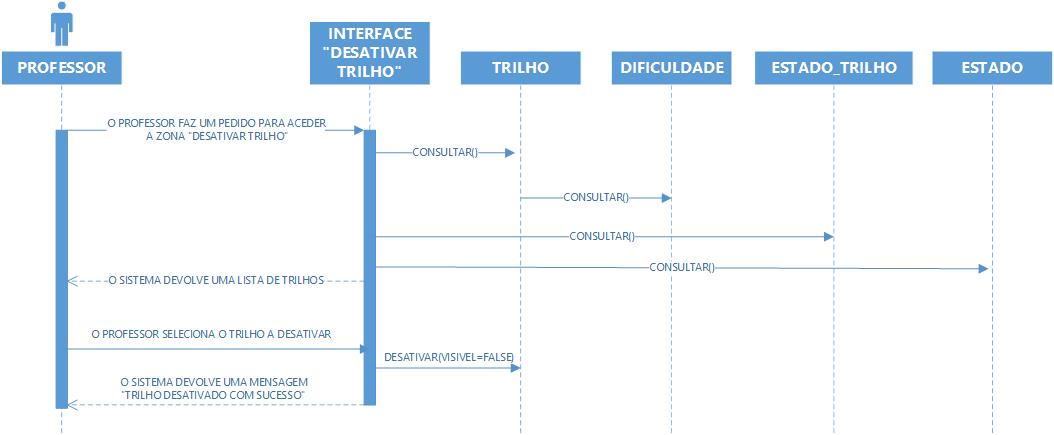


Figura 8 Diagrama de Sequencia Desativar Trilho

# Diagrama Sequencia - Consultar Trilho

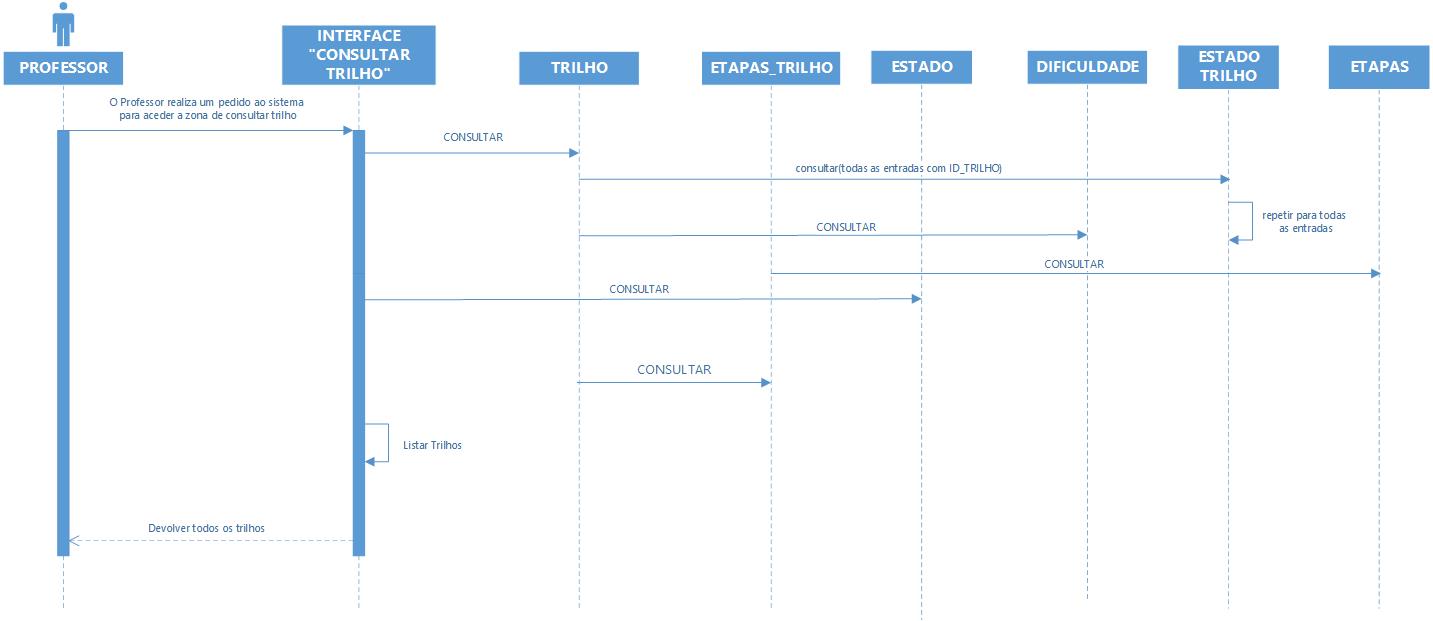


Figura 9 Diagrama de Sequencia Desativar Trilho

# Diagrama de Classes

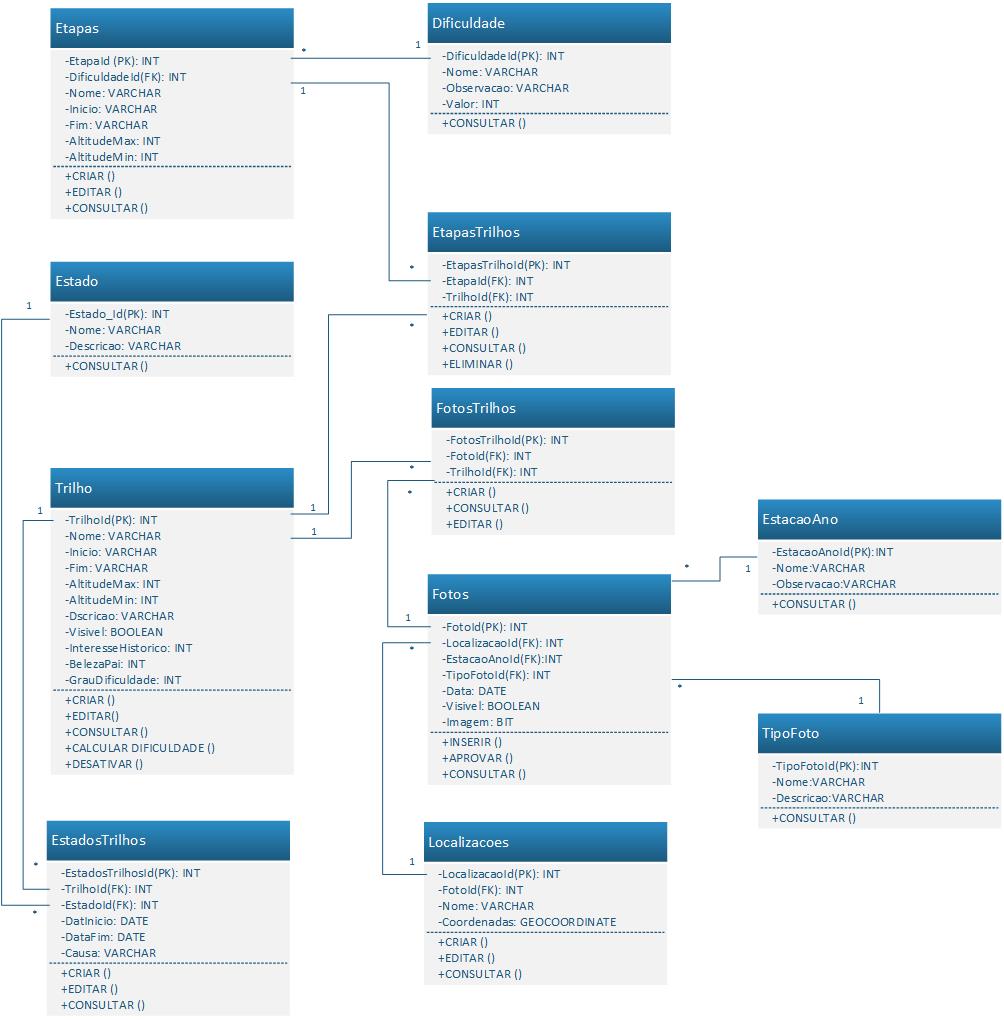


Figura 10 Diagrama de Classes

# Diagrama de Estados do Trilho



Figura 11 Diagrama de Estados

# Semântica das Classes

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Classe Estado | | | | | | | |
| **Nome do Campo** | | **Tipo de Dados** | **Descrição** | **Valores Válidos** | **Formato** | **Restrições** | **Valor por Defeito** |
| EstadoId (PK) | | INT | Número sequencial que identifica a Dificuldade (gerado automaticamente) | >0 | Até 2 digitos | Gerado pelo sistema e não pode ser alterado | - |
| Descricao | | nvarchar | Nome da Dificuldade | Caracteres de A a Z | Até 1000 Caracteres | Opcional e alterável | - |
| Nome | | nvarchar | Pequena Observação da Dificuldade | Caracteres de A a Z | De 3 a 100 caracteres | Obrigatório e alterável | - |
| **Operações** | | | | | | | |
| Nome | Descrição | | | | | | |
| Consultar() | 1. Permite a consulta dos estados possíveis. | | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Classe EtapasTrilhos | | | | | | | |
| **Nome do Campo** | | **Tipo de Dados** | **Descrição** | **Valores Válidos** | **Formato** | **Restrições** | **Valor por Defeito** |
| EtapaId (FK) | | INT | Número sequencial que identifica a Etapa | >0 | ? | Obrigatório e alterável | - |
| TrilhoId (FK) | | nvarchar | Número sequencial que identifica o Trilho | >0 | ? | Obrigatório e alterável | - |
| EtapasTrilhoId(PK) | | nvarchar | Número sequencial que identifica a linha da etapa trilho (gerado automaticamente) | >0 | ? | Gerado pelo sistema e não pode ser alterado | - |
| **Operações** | | | | | | | |
| Nome | Descrição | | | | | | |
| Criar() | Operação que permite associar várias etapas a um trilho.   1. Sistema gera o EtapasTrilhoId. 2. Sistema recebe um TrilhoId e vários EtapaId. 3. Sistema guarda na base de dados. | | | | | | |
| Editar() | Operação para adicionar ou retirar etapas a um determinado trilho.   1. Alterar as etapas pertencentes a um Trilho. | | | | | | |
| Consultar() | Operação para consultar as etapas de um determinado trilho.   1. Permite a consulta das etapas de um trilho através da consulta do Trilho. | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classe Fotos | | | | | | |
| **Nome do Campo** | **Tipo de Dados** | **Descrição** | **Valores Válidos** | **Formato** | **Restrições** | **Valor por Defeito** |
| FotoId (PK) | INT | Número sequencial que identifica a Foto | >0 | ? | Gerado pelo sistema e não pode ser alterado | - |
| Data | Datetime2(7) | Data em que foi tirada a foto | Caracteres numéricos | DD-MM-AAAA | Opcional e alterável | - |
| EstacaoAno | nvarchar | Estação do ano em que foi tirada a foto | Caracteres de A a Z | De 3 a 60 Caracteres | Opcional e alterável | - |
| LocalizacaoId(FK) | nvarchar | Número sequencial que identifica a Localização | >0 | ? | Obrigatório e alterável | - |
| TipoFotoId(FK) | INT | Número sequencial que identifica o tipo de foto | >0 | ? | Obrigatório e alterável | - |
| Imagem | BYTE | imagem | Array de bytes | imagem | Obrigatório e alterável | - |
| Visivel | Boolean | Controla a visiblilidade da imagem | logicos | True/False | Apenas alterável pelo professor | false |
| **Operações** | | | | | | |
| Nome | Descrição | | | | | |
| Inserir() | Operação de adição de uma nova foto.   1. Sistema gera o FotoId. 2. Introduzir a data. 3. Selecionar a estação do ano. 4. Selecionar o tipo de foto. 5. Introduzir localização. 6. Fazer o upload da imagem. | | | | | |
| Aprovar() | Operação que permite que a foto seja visível para todos os utilizadores   1. Um utilizador com privilégios analisa a foto e permite, ou não que a foto seja visível por todos os utilizadores. | | | | | |
| Consultar() | Permite a consulta das fotos através do FotoId. | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classe TipoFoto | | | | | | |
| **Nome do Campo** | **Tipo de Dados** | **Descrição** | **Valores Válidos** | **Formato** | **Restrições** | **Valor por Defeito** |
| TipoFotoId (PK) | INT | Número sequencial que identifica o tipo de foto | >0 | ? | Gerado pelo sistema e não pode ser alterado | - |
| Nome | nvarchar | Nome do tipo de foto (ex:fauna, flora, etc) | Caracteres de A a Z | Até 1000 Caracteres | Obrigatório e alterável | - |
| Descricao | nvarchar | Pequena descrição do tipo de foto | Caracteres de A a Z | De 0 a 1000 Caracteres | Opcional e alterável | - |
| **Operações** | | | | | | |
| Nome | Descrição | | | | | |
| Consultar() | Permite a consulta dos tipos de fotos possíveis. | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classe Dificuldade | | | | | | |
| **Nome do Campo** | **Tipo de Dados** | **Descrição** | **Valores Válidos** | **Formato** | **Restrições** | **Valor por Defeito** |
| DificuldadeId (PK) | Int | Número sequencial que identifica a Dificuldade (gerado automaticamente) | >0 | ? | Gerado pelo sistema e não pode ser alterado | - |
| Nome | nvarchar | Nome da Dificuldade | Caracteres de A a Z | De 3 a 100 Caracteres | Obrigatório e alterável | - |
| Observacao | nvarchar | Pequena Observação da Dificuldade | Todos os caracteres | Até 250 Caracteres | Opcional e alterável | - |
| Valor | Int | Valor que classifica a dificuldade | Inteiro de 1 a 5 | 1 Catacteres | Obrigatório e alterável | - |
| **Operações** | | | | | | |
| Nome | Descrição | | | | | |
| Consultar() | Permite a consulta dos graus de dificuldade existentes. | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classe EstadosTrilhos | | | | | | |
| **Nome do Campo** | **Tipo de Dados** | **Descrição** | **Valores Válidos** | **Formato** | **Restrições** | **Valor por Defeito** |
| EstadosTrilhosId (FK) | Int | Chave primaria | >0 | ? | Gerado pelo sistema e não pode ser alterado | - |
| TrilhoId (FK) | Int | Chave estrangeira que identifica um Trilho | >0 | ? | Obrigatório e alterável | - |
| EstadoId (Fk) | Int | Chave estrangeira que identifica um Estado | >0 | ? | Obrigatório e alterável | - |
| Causa | nvarchar | Motivo do qual resulta o Estado do trilho | Todos os caracteres | De 3 a 100 Caracteres | Opcional e alterável | - |
| DataInicio | datetime2(7) | Data em que inicia o periodo em que o estado esta activo | Caracteres numéricos | DD-MM-AAAA | Opcional e alterável | - |
| DataFim | datetime2(7) | Data em que termina o periodo em que o estado esta activo | Caracteres numéricos | DD-MM-AAAA | Opcional e alterável | - |
| **Operações** | | | | | | |
| Nome | Descrição | | | | | |
| Criar() | Operação que permite associar vários trilhos a um estado.   1. Sistema gera o EstadosTrilhosId. 2. Sistema recebe um TrilhoId e vários EstadoId. 3. Sistema guarda na base de dados. | | | | | |
| Editar() | Operação que permite adicionar ou retirar trilhos de um o estado.   1. Alterar o estado de um trilho. | | | | | |
| Consultar() | Permite a consulta do estado de um trilho. | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classe Etapas | | | | | | |
| **Nome do Campo** | **Tipo de Dados** | **Descrição** | **Valores Válidos** | **Formato** | **Restrições** | **Valor por Defeito** |
| EtapaId (PK) | Int | Número sequencial que identifica a Etapa (gerado automaticamente) | >0 | ? | Gerado pelo sistema e não pode ser alterado | - |
| AltitudeMax | Int | Altitude máxima da etapa | <2500  >0 | Numero de 0 a 2500 | Opcional e alterável | - |
| AltitudeMin | Int | Altitude minima da etapa | <2500  >0 | Numero de 0 a 2500 | Opcional e alterável | - |
| DificuldadeId | Int | Chave estrangeira que identifica uma Dficuldade | >0 | ? | Obrigatório e alterável | - |
| Fim | nvarchar | Local onde termina a etapa | Todos os Caracteres | De 3 a 100 Caracteres | Obrigatório e alterável | - |
| Inicio | nvarchar | Local onde inicia a etapa | Todos os Caracteres | 3 a 100 Caracteres | Obrigatório e alterável | - |
| Nome | nvarchar | Nome da etapa | Todos os Caracteres | De 3 a 100 Caracteres | Obrigatório e alterável | - |
| **Operações** | | | | | | |
| Nome | Descrição | | | | | |
| Criar() | Operação que permite criar uma nova etapa.   1. Sistema gera EtapaId. 2. Selecionar Altitude máxima. 3. Selecionar Altitude mínima. 4. Selecionar Dificuldade. 5. Introduzir Início. 6. Introduzir Fim. 7. Introduzir Nome. 8. Criar Etapa. | | | | | |
| Editar() | Operação que permite editar os dados de uma etapa.   1. Alterar EtapaId. 2. Alterar Altitude máxima. 3. Alterar Altitude mínima. 4. Alterar Início. 5. Alterar Fim. 6. Alterar Nome. | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Classe FotosTrilhos | | | | | | | |
| **Nome do Campo** | **Tipo de Dados** | **Descrição** | | **Valores Válidos** | **Formato** | **Restrições** | **Valor por Defeito** |
| FotoId (FK) | Int | Chave estrangeira que identifica uma Foto | | >0 | ? | Obrigatório e alterável | - |
| TrilhoId (FK) | Int | Chave estrangeira que identifica um trilho | | >0 | ? | Obrigatório e alterável | - |
| FotosTrilhoId (PK) | Int | Número sequencial que identifica um objecto FotosTrilho (gerado automaticamente) | | >0 | ? | Gerado pelo sistema e não pode ser alterado | - |
| **Operações** | | | | | | | |
| Nome | Descrição | | | | | | |
| Criar() | Operação que associa Fotos a um determinado trilho, é executada quando o utilizador cria um trilho.   1. Sistema gera o FotosTrilhosId. 2. Sistema recebe um TrilhoId e vários FotoId. 3. Sistema guarda na base de dados. | | | | | | |
| Editar() | Operação que permite adicionar ou retirar fotos de um o trilho.   1. Alterar as fotos de um trilho | | | | | | |
| Consultar() | Permite a consulta das fotos de um trilho. | | | | | | |
| Classe Localizações | | | | | | | |
| **Nome do Campo** | **Tipo de Dados** | **Descrição** | **Valores Válidos** | | **Formato** | **Restrições** | **Valor por Defeito** |
| LocalizacaoId (PK) | Int | Número sequencial que identifica a Localização (gerado automaticamente) | >0 | | ? | Gerado pelo sistema e não pode ser alterado | - |
| Nome | nvarchar | Nome da Localização | Todos os Caracteres | | Até 60 Caracteres | Obrigatório e alterável | - |
| Coordenadas | String | Coordenadas da Localização | ? | | GeoCoordinate | Opcional e Alterável | - |
| **Operações** | | | | | | | |
| Nome | Descrição | | | | | | |
| Consultar() | Operação que permite consultar a localização de uma foto. | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classe Trilho | | | | | | |
| **Nome do Campo** | **Tipo de Dados** | **Descrição** | **Valores Válidos** | **Formato** | **Restrições** | **Valor por Defeito** |
| TrilhoId (PK) | Int | Número sequencial que identifica um trilho (gerado automaticamente) | >0 | ? | Gerado pelo sistema e não pode ser alterado | - |
| AltitudeMax | Int | Altitude máxima do trilho | <2500  >0 | Numero de 0 a 2500 | Opcional e alterável | - |
| AltitudeMin | Int | Altitude máxima da etapa | <2500  >0 | Numero de 0 a 2500 | Opcional e alterável | - |
| BelezaPaisagistica | Int | Valor que classifica a beleza da foto | >0 | Numero entre 1 e 5 | Obrigatório e alterável | - |
| Interessehistorico | Int | Valor que classifica o interesse histórico da foto | >0 | Numero entre 1 e 5 | Obrigatório e alterável | - |
| Descricao | nvarchar | Pequena Observação da Dificuldade | Todos os caracteres | Até 1000 Caracteres | Opcional e alterável | - |
| Fim | nvarchar | Local onde termina o trilho | Todos os Caracteres | Até 60 Caracteres | Obrigatório e alterável | - |
| Inicio | nvarchar | Local onde inicia o trilho | Todos os Caracteres | Até 60 Caracteres | Obrigatório e alterável | - |
| Nome | nvarchar | Nome do Trilho | Todos os Caracteres | Até 60 Caracteres | Obrigatório e alterável | - |
| Visivel | bit | Eliminação lógica da base de dados | Booleano | ? | Opcional e alterável | True |
| GrauDificuldade | INT | Grau de dificuldade calculado através da media de dificuldades das etapas | >0 | Numero entre 1 e 5 | Calculado pela média da dificuldade das etapas, não pode ser inserido ou alterado | 0 |
| **Operações** | | | | | | |
| Nome | Descrição | | | | | |
| Criar() | Operação que permite a criação de um novo trilho.  Operação que permite criar uma nova etapa.   1. Sistema gera TrilhoId. 2. Selecionar Altitude máxima. 3. Selecionar Altitude mínima. 4. Selecionar Beleza paisagística. 5. Selecionar Interesse Histórico. 6. Introduzir Início. 7. Introduzir Fim. 8. Introduzir Nome. 9. Introduzir Descrição. 10. Introduzir Etapas. 11. Criar Trilho. | | | | | |
| Editar() | Operação de edição de um trilho.   1. Alterar Altitude máxima. 2. Alterar Altitude mínima. 3. Alterar Beleza paisagística. 4. Alterar Interesse Histórico. 5. Alterar Início. 6. Alterar Fim. 7. Alterar Nome. 8. Alterar Descrição. 9. Alterar Etapas. | | | | | |
| Consultar() | 1. Permite a consulta de um trilho através do TrilhoId. | | | | | |
| Calcular Dificuldade() | Operação que calcula a dificuldade de um trilho através do calculo da média das dificuldades das etapas. É chamada quando o utilizador cria ou edita um trilho. | | | | | |
| Desativar() | Operação de alteração da variável “Visível” do trilho, se este está visível aos utilizadores. | | | | | |

# Diagrama de Atividades

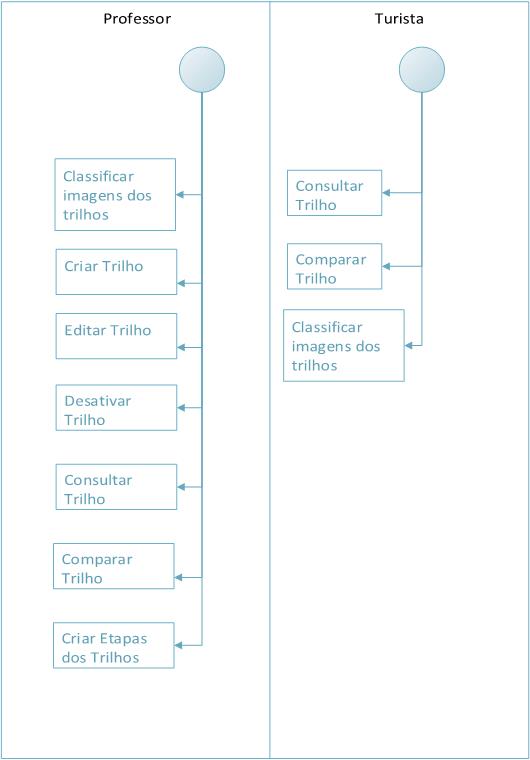


Figura 12 Diagrama de Atividades

# Diagrama de Componentes

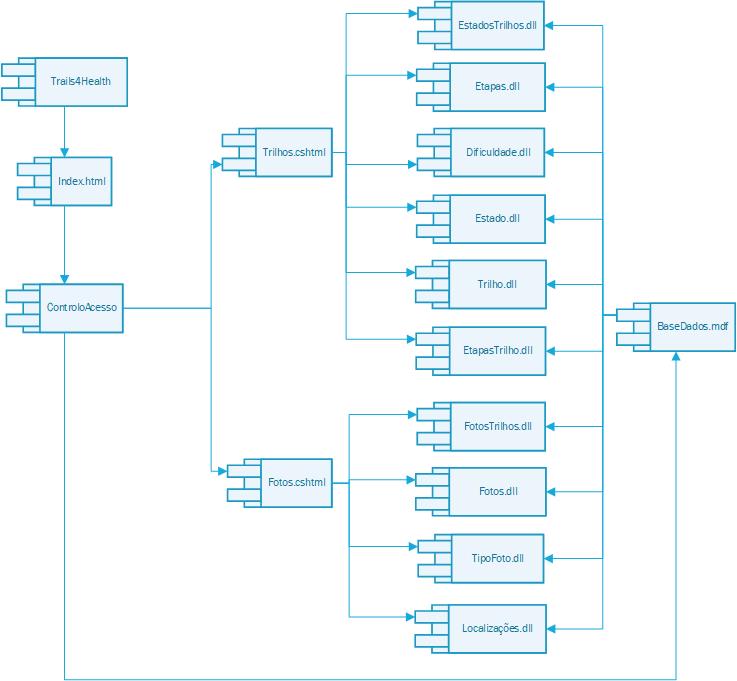


Figura 13 Diagrama de Componentes

# Diagrama de Pacotes de Casos de Uso



Figura 14 Diagrama de Pacotes de casos de uso

## Diagrama de Casos de uso do Pacote Professor

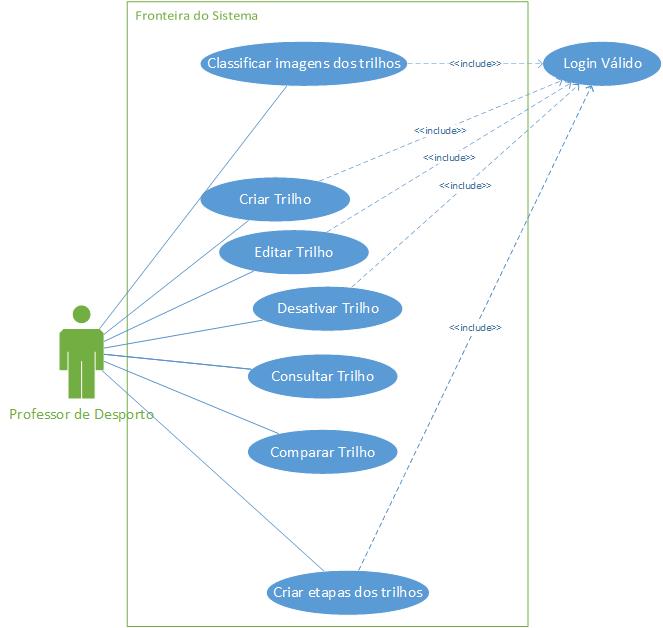


Figura 15 Diagrama de Casos de Uso do Professor

## Diagrama de Casos de uso do Pacote Turista

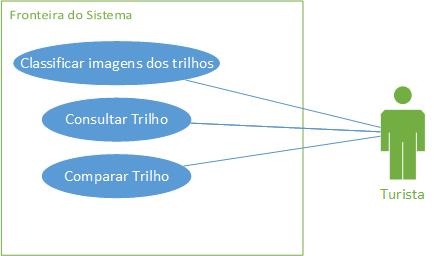


Figura 16 Diagrama de Casos de Uso do Turista

# Diagrama de Instalação

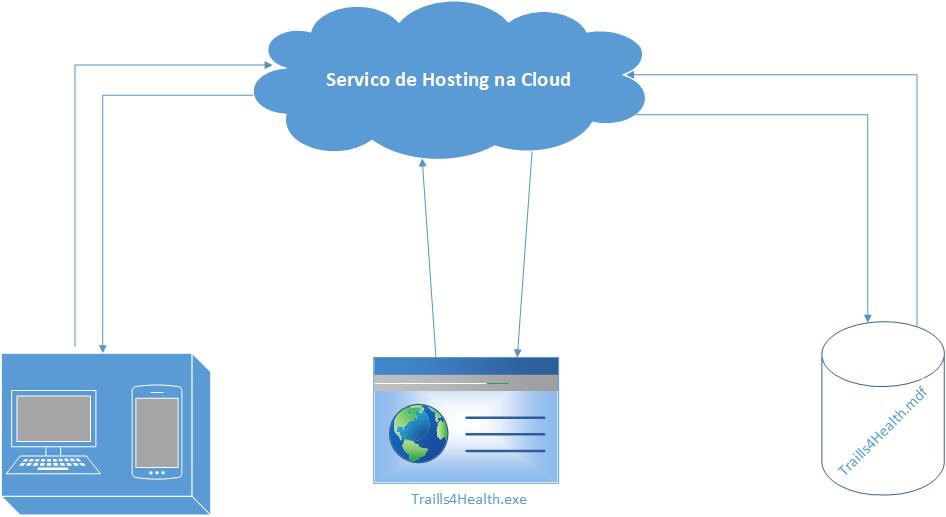


Figura 17 Diagrama de Instalação

# Print screen das tabelas da Base de dados

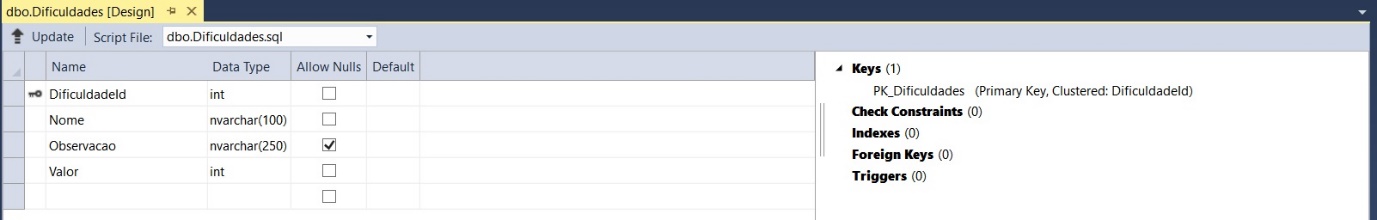


Figura 18 Tabela Dificuldades (BaseDados)

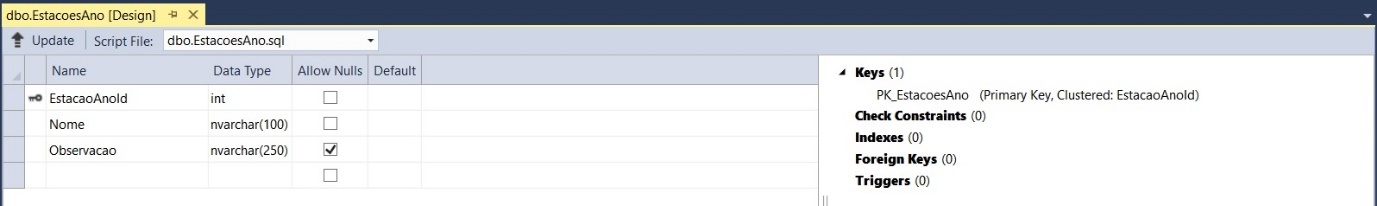


Figura 19 Tabela EstacoesAno (BaseDados)

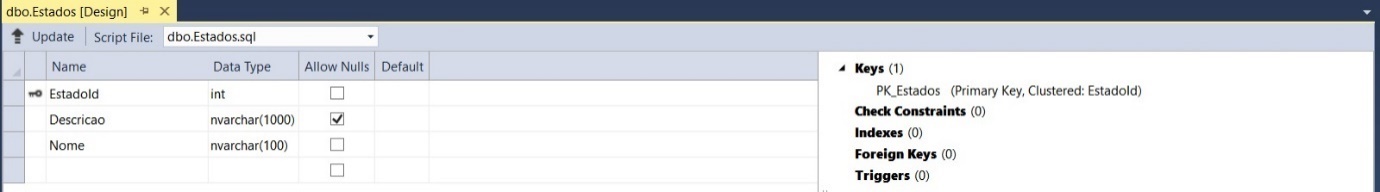


Figura 20 Tabela Estados (BaseDados)

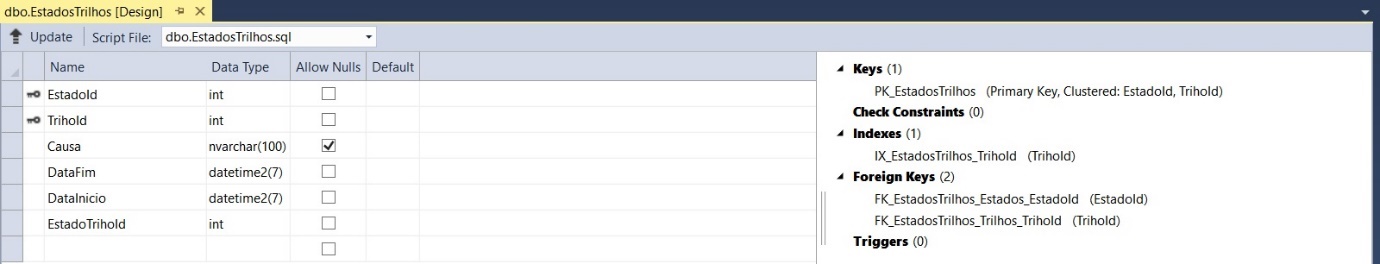


Figura 21 Tabela EstadosTrilhos (BaseDados)

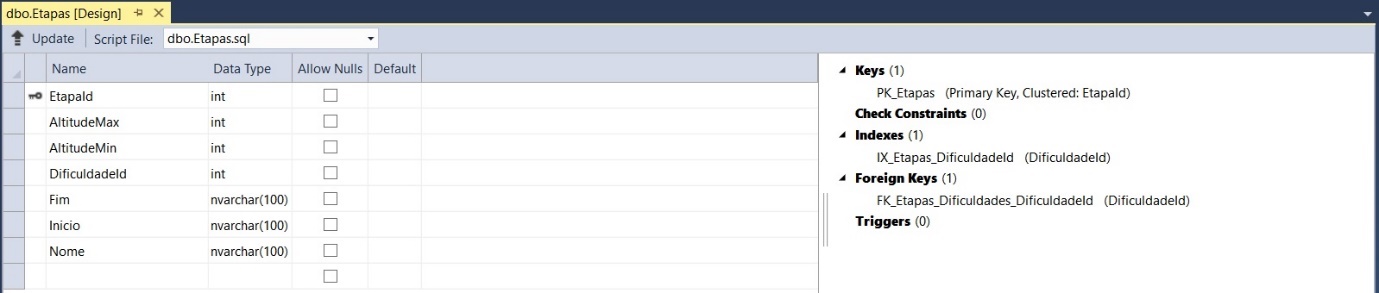


Figura 22 Tabela Etapas (BaseDados)

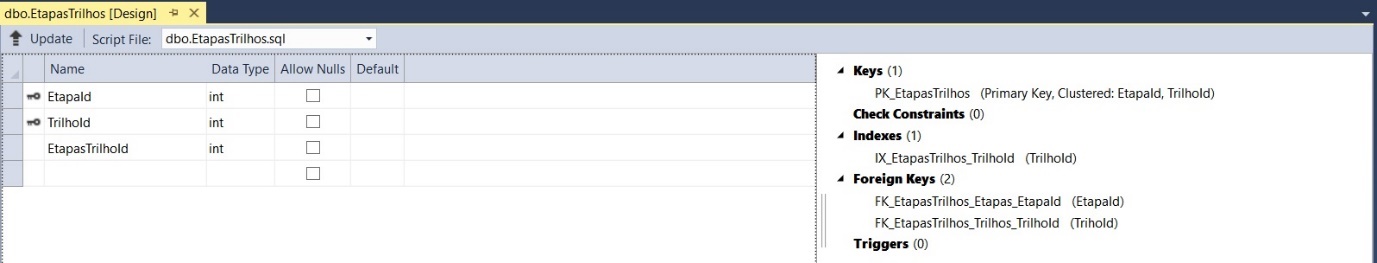


Figura 23 Tabela EtapasTrilhos (BaseDados)

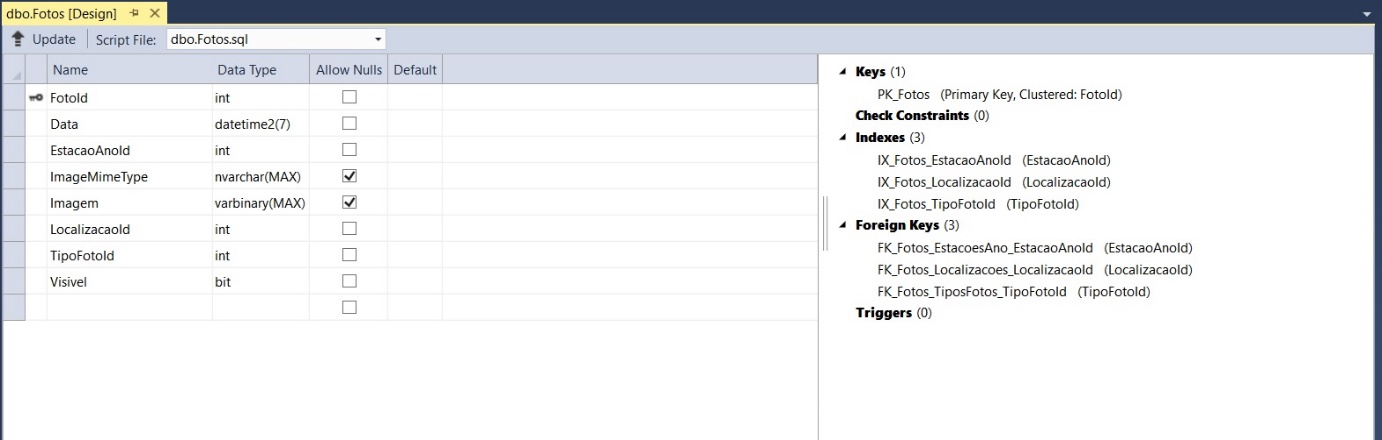


Figura 24 Tabela Fotos (BaseDados)

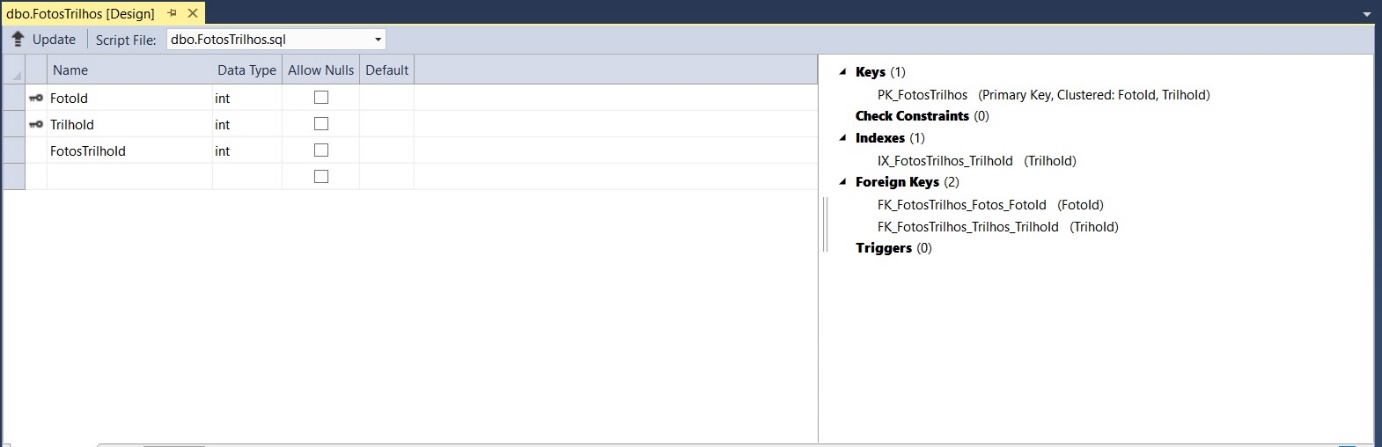


Figura 25 Tabela Trilhos (BaseDados)

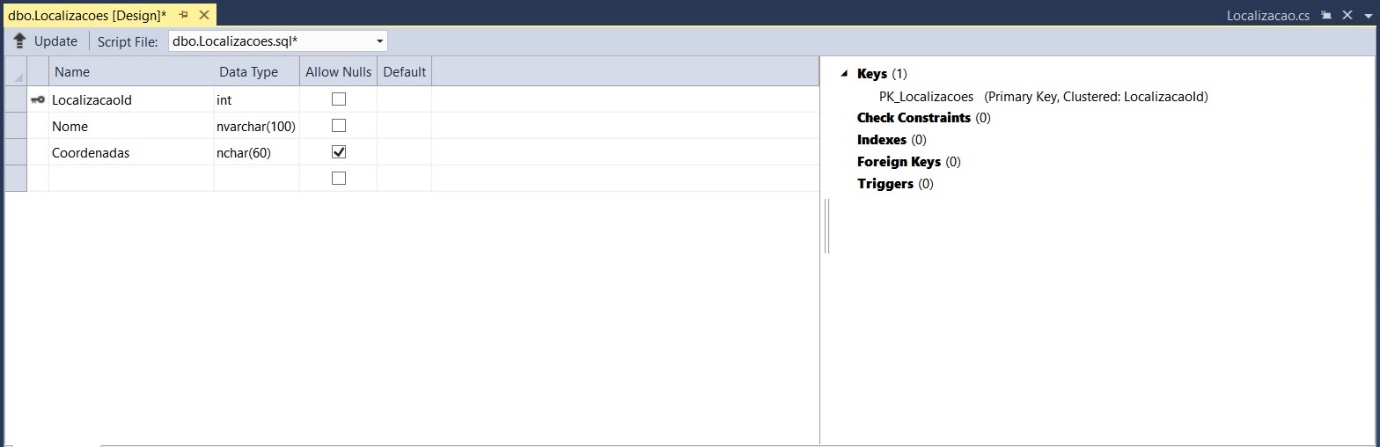


Figura 26 Tabela Localizacao (BaseDados)

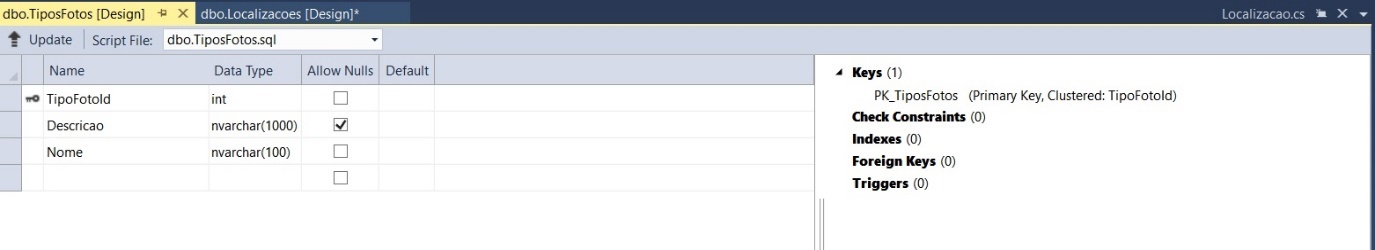


Figura 27 Tabela TiposFotos (BaseDados)

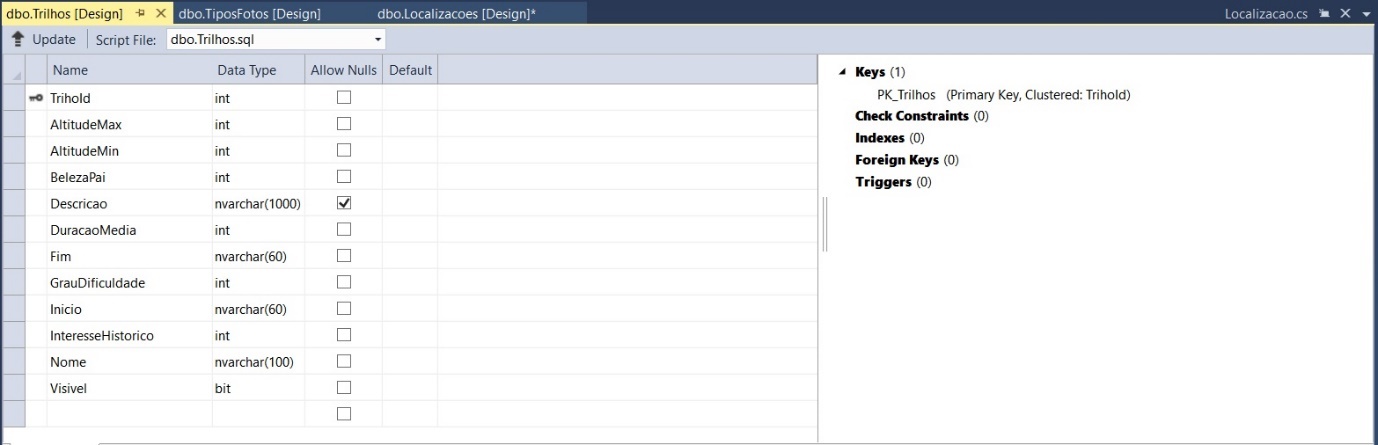


Figura 28 Tabela Trilhos (BaseDados)

# Print screen dos dados da Base de dados

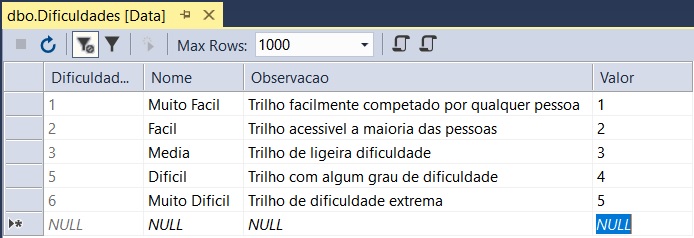


Figura 29 Tabela Dificuldades (BaseDados Preenchida)

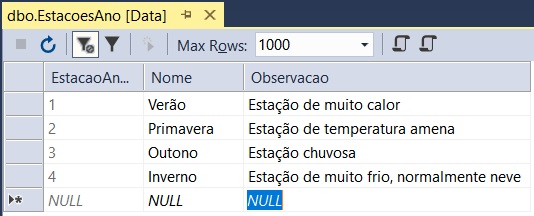


Figura 30Tabela EstacoesAno (BaseDados Preenchida)

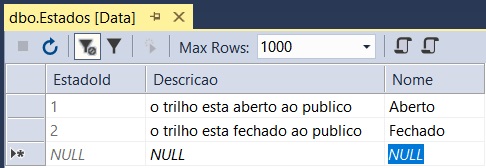


Figura 31Tabela Estados (BaseDados Preenchida)

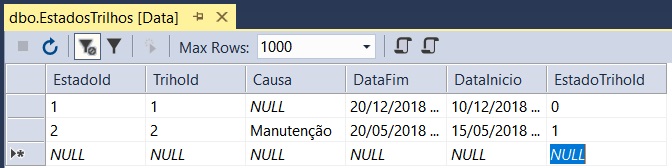


Figura 32Tabela EstadosTrilhos (BaseDados Preenchida)



Figura 33 Tabela Etapas (BaseDados Preenchida)

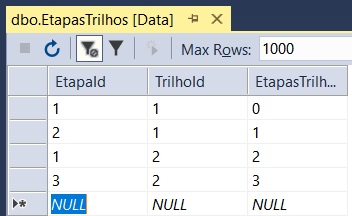


Figura 34 Tabela EtapasTrilhos (BaseDados Preenchida)

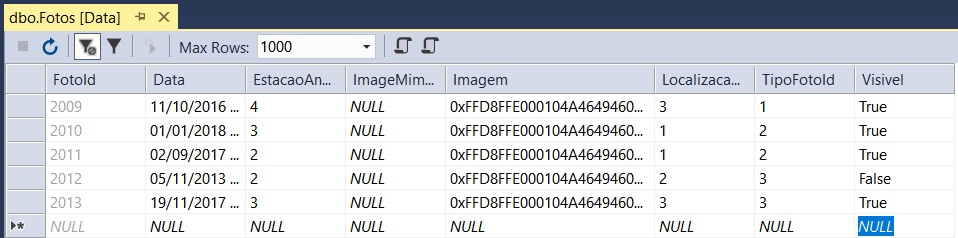


Figura 35 Tabela Fotos (BaseDados Preenchida)

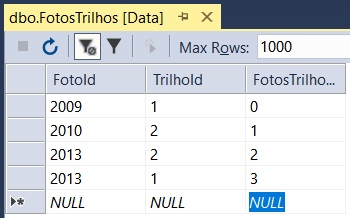


Figura 36 Tabela FotosTrilhos (BaseDados Preenchida)

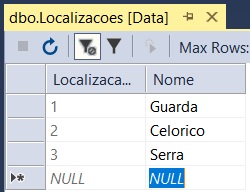


Figura 37 Tabela Localizacoes (BaseDados Preenchida)

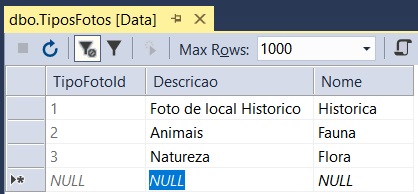


Figura 38 Tabela TiposFotos (BaseDados Preenchida)

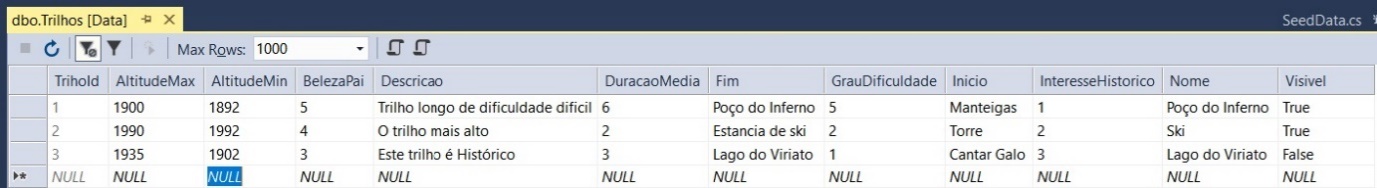


Figura 39 Tabela Trilhos (BaseDados Preenchida)

# Protótipos da aplicação



Figura 40 Index (Pagina Inicial)

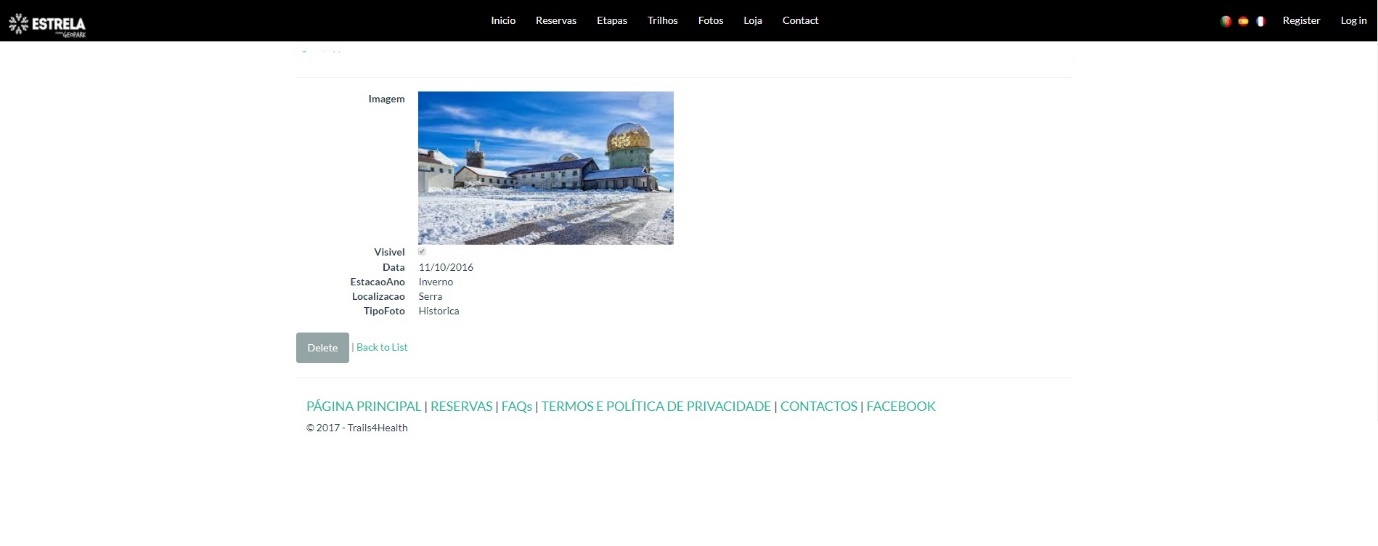


Figura 41Eliminar Fotos

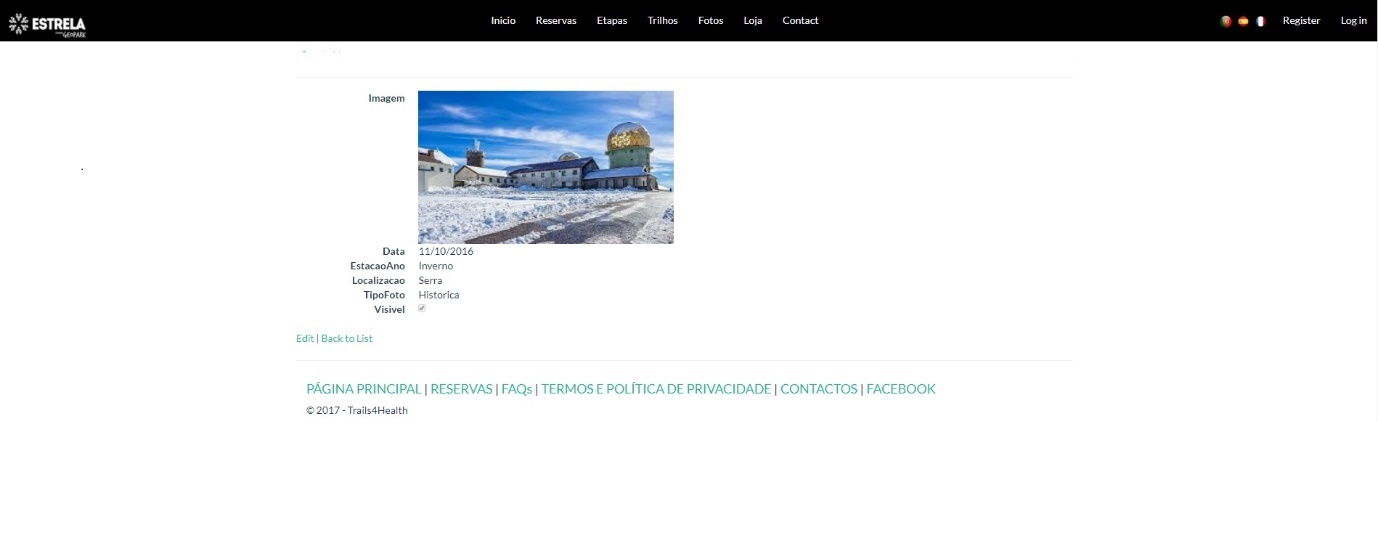


Figura 42 Detalhes Foto

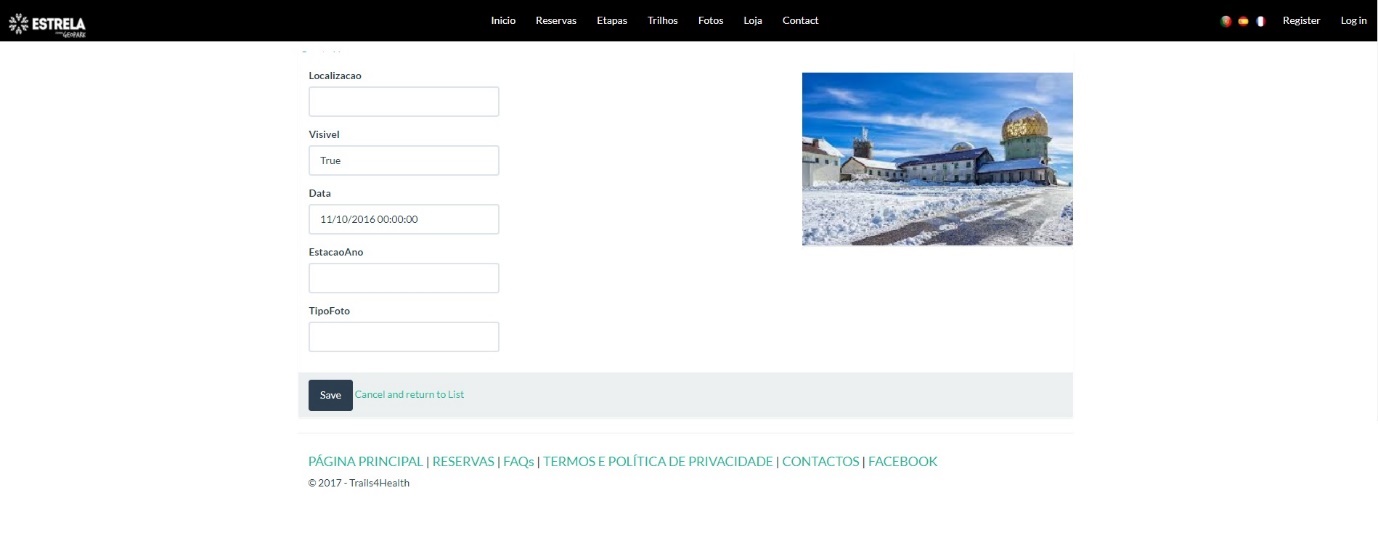


Figura 43 Editar Foto

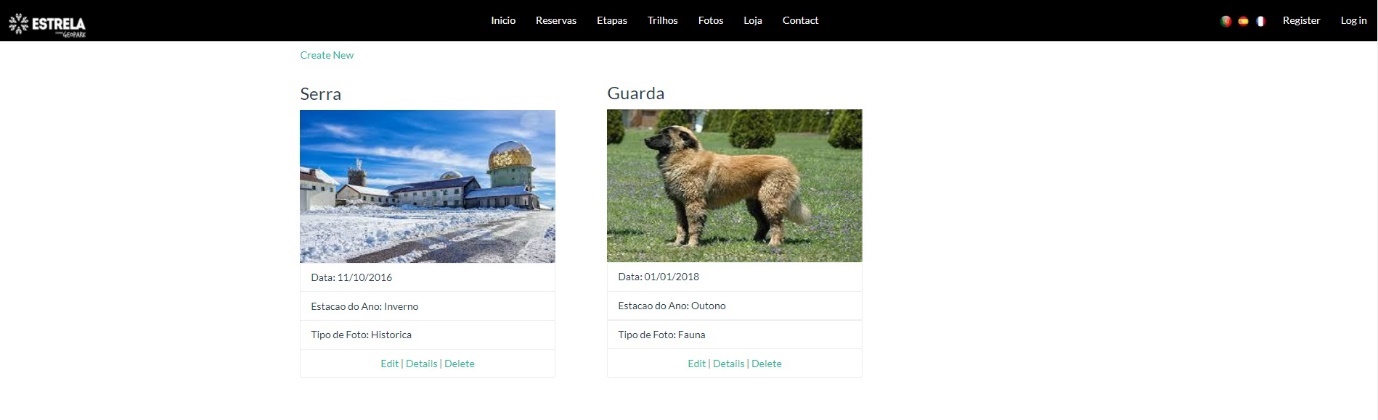


Figura 44 Index Fotos



Figura 45 Index Trilho

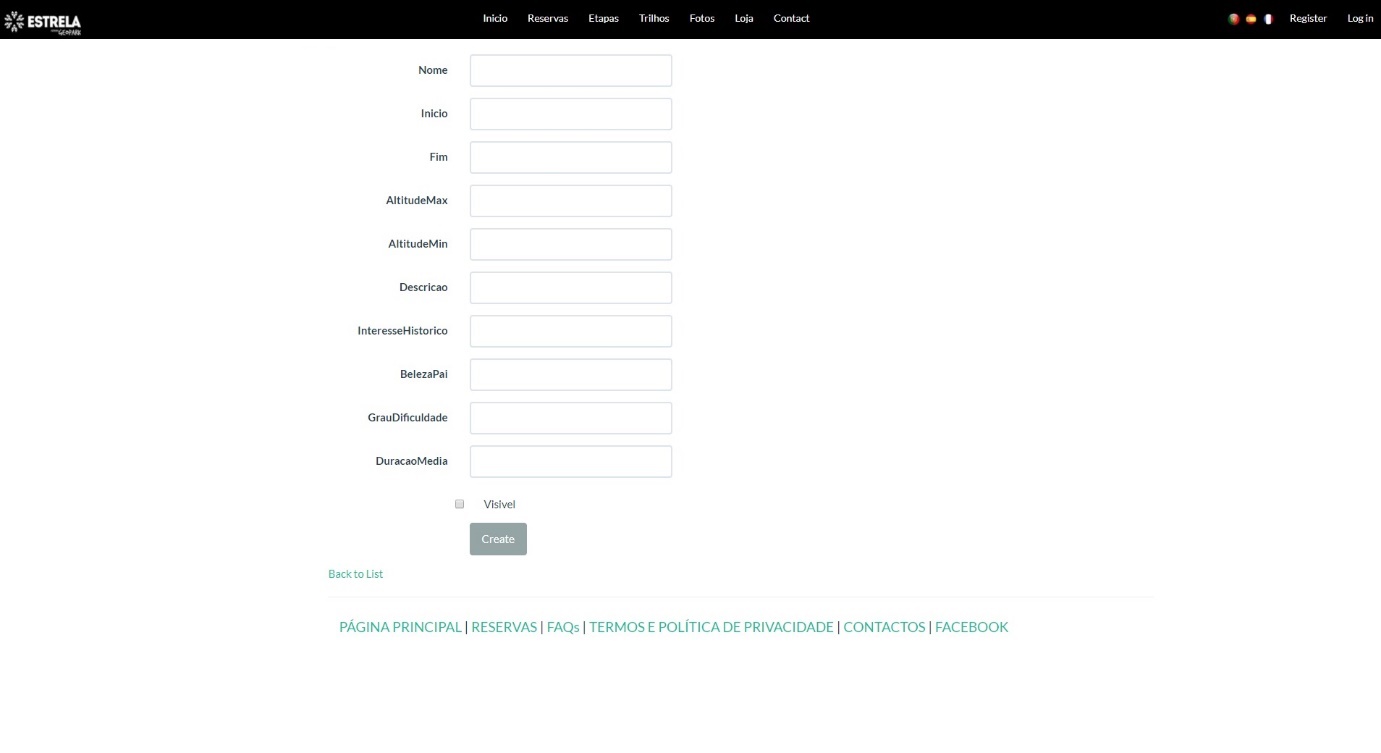


Figura 46 Criar Trilho

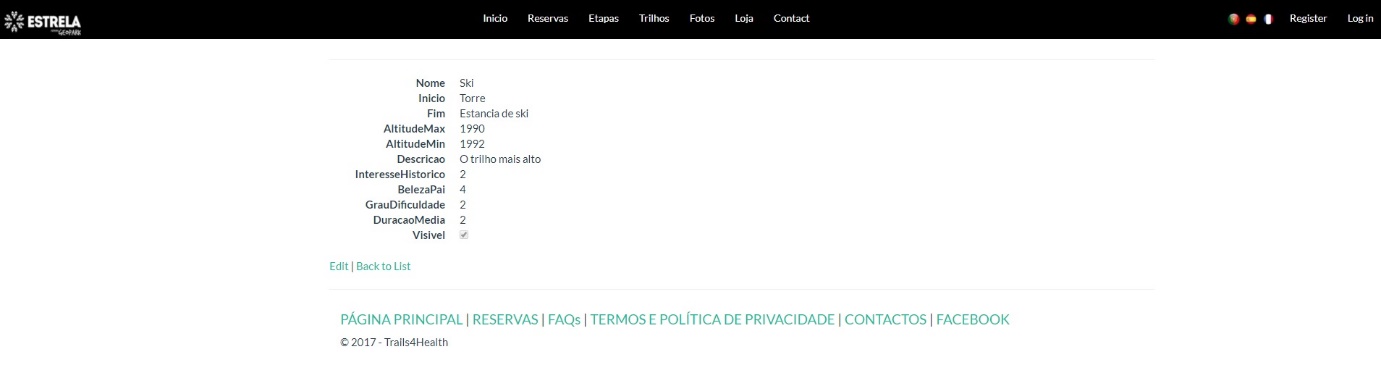


Figura 47 Consultar Trilho

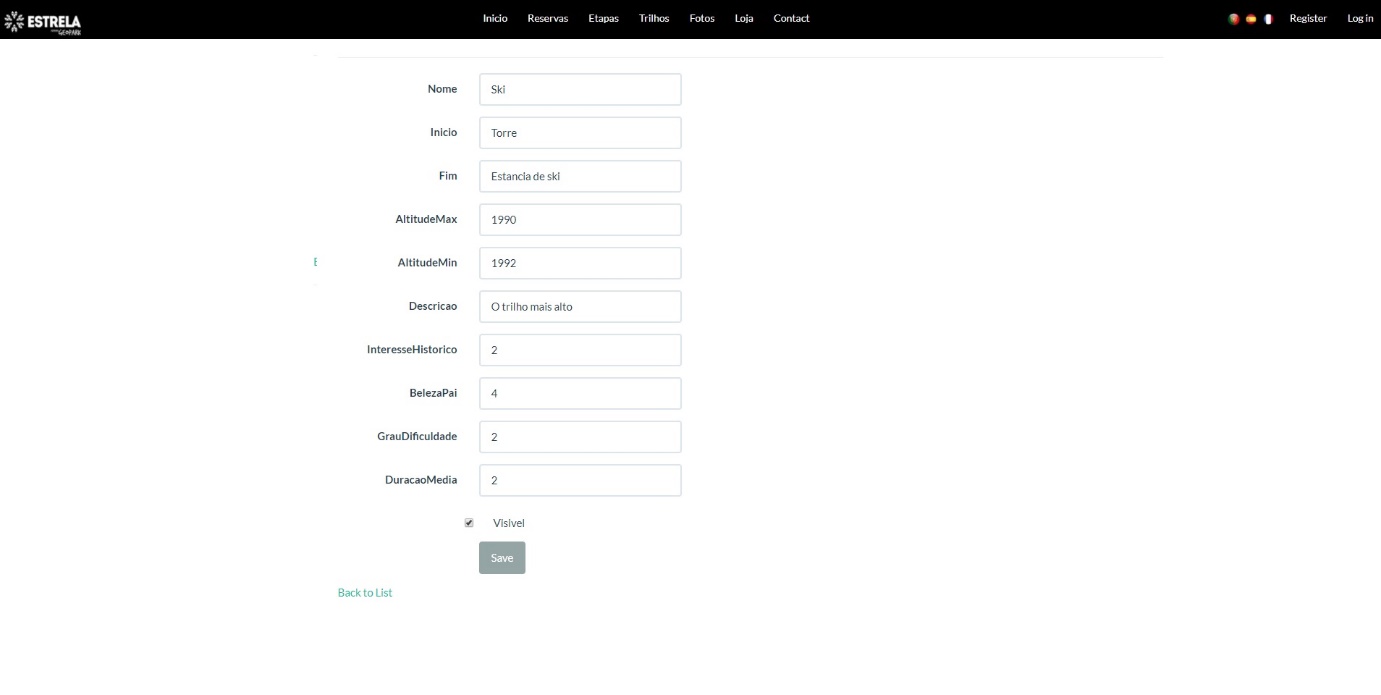


Figura 48 Editar Trilho



Figura 49 Comparar Trilho

# Conclusão

Este projeto ajudou-nos a desenvolver competências na engenharia de software, como começar e projetar uma aplicação, bem como desenvolvê-la numa linguagem de alto nível, recorrendo a ferramentas que muito provavelmente iremos utilizar na nossa vida profissional, como o GitHub, ASP.NET MVC, Microsoft Visio, Visual Studio, Excel, entre outros. O desenvolvimento não foi fácil, deparando-nos variadas vezes com problemas, dos quais não pensamos conseguir ultrapassar. Apesar dos altos e baixos, conseguimos concluir o projeto com sucesso, resolvendo os problemas muitas vezes com recurso aos diagramas XML, e reuniões com os docentes. Classificamos, por isto, este trabalho prático como extremamente enriquecedor para as nossas competências pessoais e profissionais.

# Anexos