



Universidade do Minho

Escola de Engenharia

Licenciatura em Engenharia informática

Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Lectivo de 2021/2022

Colere

**Cláudia Silva, Daniela Carvalho, Eduardo
Magalhães, Laura Rodrigues, Mariana Rodrigues**

Novembro 2021

Data de Recepção	
Responsável	
Avaliação	
Observações	

Colere

**Cláudia Silva, Daniela Carvalho, Eduardo
Magalhães, Laura Rodrigues, Mariana Rodrigues**

Novembro 2021

Resumo

O seguinte relatório foi elaborado no âmbito de uma proposta de desenvolvimento de um guia de locais de interesse, mais concretamente museus e teatros para a cidade de Braga.

Numa fase inicial, após decidir um tópico do trabalho em concreto e contextualizar o problema, fizemos toda a pesquisa que consideramos necessária para depois desenvolver e materializar o mesmo.

Após a fase inicial, procedeu-se à análise de requisitos. Para essa análise ser sólida, definimos estratégias como fazer reuniões com as entidades culturais e recolhemos respostas de alguns inquéritos à população em geral. Assim, conseguimos recolher e debater todos os requisitos funcionais e não funcionais que os mesmos pretendiam e que, posteriormente, a nossa aplicação vai implementar.

De seguida, através do desenvolvimento de alguns diagramas UML, de modo a representar de uma forma abstrata as funcionalidades que o sistema deve implementar para cumprir todos os requisitos recolhidos e validados na fase anterior.

Com vista em armazenar e manter os dados que achamos ser essenciais para o desenvolvimento do projeto íntegros, desenvolvemos um modelo lógico da base de dados que pretendemos implementar que, numa fase posterior, vai servir de base para a implementação física da mesma.

Por último, idealizamos a maneira como toda a informação vai ser apresentada aos diversos tipos de utilizador esboçando um modelo de interfaces do sistema. Uma vez idealizado e concluído o esboço das interfaces, apresentamos o resultado do mesmo a todas as entidades envolvidas e interessadas no projeto que se mostraram agradadas com o modelo de interfaces proposto.

Concluindo, com a especificação e fundamentação do projeto conseguimos pensar de maneira mais detalhada no sistema de software a desenvolver, assim como antecipar possíveis imperfeições na implementação do mesmo.

Área de Aplicação: Cultura

Palavras-Chave: Engenharia de Software, Colere, Cultura, Museus, Teatros, Eventos, Percursos

Índice

1. Introdução	1
1.1. Contextualização	1
1.2. Apresentação do Caso de Estudo	Erro!
Marcador não definido.	
1.3. Motivação e Objectivos	Erro!
Marcador não definido.	
1.4. Estrutura do Relatório	Erro!
Marcador não definido.	
2. Sugestões para Escrita do Relatório	Erro!
Marcador não definido.	
2.1. Sugestões Gerais	Erro!
Marcador não definido.	
2.2. Termos Estrangeiros	Erro!
Marcador não definido.	
2.3. Tabelas e Figuras	Erro!
Marcador não definido.	
2.4. Siglas e Acrónimos	Erro!
Marcador não definido.	
2.5. Referências Bibliográficas	Erro!
Marcador não definido.	
2.6. Tipo de Ficheiro	Erro!
Marcador não definido.	
3. Conclusões e Trabalho Futuro	11
Bibliografia	Erro!
Marcador não definido.	
Referências WWW	Erro!
Marcador não definido.	
Lista de Siglas e Acrónimos	Erro!
Marcador não definido.	

Índice de Figuras

Figura 1 - Logotipo da aplicação	2
Figura 2 - Maquete	4
Figura 3 - Gantt parte1	6
Figura 4 - Gantt parte 2	6
Figura 6 - Modelo de Domínios	12
Figura 7 - Diagrama de Use Cases	17
Figura 8 - Diagrama de sequências 1	18
Figura 9 - Diagrama de sequências 2	19
Figura 10 - Diagrama de sequências 3	20
Figura 11 - Diagrama de sequências 4	20
Figura 12 - Diagrama de sequências 5	21
Figura 13 - Diagrama de sequências 6	22
Figura 14 - Diagrama de sequências 7	22
Figura 15 - Diagrama de sequências 8	23
Figura 16 - Diagrama de sequências 9	23
Figura 17 - Diagrama de sequências 10	24
Figura 18 - Diagrama de Classes	25
Figura 19 - Diagrama de atividades 1	26
Figura 20 - Diagrama de atividades 2	27
Figura 21 - Diagrama de atividades 3	28
Figura 22 - Diagrama de atividades 4	28
Figura 23 - Diagrama de atividades 5	29
Figura 24 - Diagrama de atividades 6	29
Figura 25 - Diagramas de atividades 7	30
Figura 26 - Diagrama de atividades 8	30
Figura 27 - Diagrama de atividades 9	31
Figura 28 - Modelo lógico da Base de Dados	33
Figura 29 - Mockups	34

1. Introdução

1.1. Contextualização

Um dos setores mais importantes em qualquer sociedade é o setor da cultura. Cultura é muito mais que mero entretenimento e cada museu, livro e espetáculo nos conta uma história diferente, ajuda a cultivar o intelecto e muitas das vezes abre os olhos ao povo criticando o que de mal se passa na sociedade.

"A cultura assusta muito. (...) Um povo que lê, nunca será um povo de escravos.", António Lobo Antunes.

Como sabemos, a situação pandémica em que vivemos prejudicou consideravelmente o setor da cultura no nosso país. Durante os confinamentos vividos, fecharam museus, teatros, salas de cinema, não se realizaram espetáculos nem qualquer outro evento cultural. Não sendo gerada receita, muitos artistas, companhias de teatro e famílias que vivem da cultura passaram por muitas dificuldades, tendo algumas ficado sem qualquer rendimento.

Posto isto, depois de reunirmos e debatermos sobre o assunto, pensamos que seria interessante desenvolver um software que desse a conhecer aos seus utilizadores pontos de interesse e eventos culturais em Braga com o intuito de ajudar os agentes culturais da região através da divulgação e assim impulsionar um aumento nos seus rendimentos. Com o aumento significativo das tecnologias de informação e comunicação ao longo dos anos, este software poderia ser uma mais valia, não apenas na divulgação de locais de interesse cultural tais como museus e teatros, mas também no que toca à formação pessoal de qualquer indivíduo.

1.2. Motivação e Objetivos

Tal como referido anteriormente, o impacto da pandemia foi devastador para a economia. Decidimos incentivar a mudança de modo a impedir que se repita o que tem acontecido até então. Acreditamos que desenvolver uma aplicação que divulgue locais de interesse e eventos culturais de modo a reduzir o impacto que a pandemia teve em empresas dedicadas à cultura, museus e outras entidades culturais.

Com a aplicação pretendemos atrair mais atenção para estes locais e, consequentemente, aumentar a faturação dos mesmos, o que ajudará a impedir que mais empresas declarem falência.

Temos também como objetivo dar a conhecer a cidade de Braga facilitando o acesso e a divulgação de locais de interesse como museus e teatros (monumentos) e anunciar atempadamente eventos culturais permitindo que mais pessoas compareçam e participem nestes.

1.3. Justificação e utilidade do sistema

Infelizmente, dada a situação pandémica em que vivemos há diversos negócios e empresas que vão sendo obrigados a fechar devido às várias dificuldades económicas que estão a enfrentar.

Desta forma, no software desenvolvido apresentamos locais como museus, teatros e eventos culturais com vista a promover a economia local. Para além disso, esperamos atrair mais turistas para Braga, dado que, a aplicação auxilia outros setores também.

Consideramos também, que há poucas informações (como páginas na internet , entre outras) que dinamizam o turismo na zona de Braga. Desta forma, a nossa aplicação é pioneira nesta região.

1.4. Estabelecimento da identidade do projeto



Figura 1 - Logotipo da aplicação

Nome: Colere

Slogan: Cuidar, Cultivar e Crescer

Idioma: Português

Categoria: Cultura, Viagem

Faixa Etária: PEGI 3

Descrição: Uma aplicação que divulga locais de índole cultural e eventos que ocorrem nos mesmos.

Criadores: Cláudia Silva, Daniela Carvalho, Eduardo Magalhães, Laura Rodrigues e Mariana Rodrigues

1.5. Identificação dos recursos necessários

Para garantir uma boa prática no desenvolvimento de aplicações é fundamental que se assegurem todos os recursos necessários, uma vez que sem estes estaríamos a comprometer a sua implementação e funcionamento. Assim, foi importante para nós reunir e debater sobre as necessidades que a aplicação teria.

De modo a garantir que todas as especificações da aplicação são desenvolvidas, definimos que serão necessários 2/3 programadores. Dada a complexidade desta aplicação teremos 1 Engenheiro de Software que garanta assim a gestão das tarefas e o bom funcionamento da equipa e da aplicação.

Estimamos ainda que será necessário 1 designer que fique encarregue de dar à aplicação uma vista agradável e garantir a simplicidade de modo a facilitar a sua utilização e tornando perceptível a seleção das várias funcionalidades para qualquer utilizador.

1.6. Maqueta do sistema

A maqueta desenvolvida permite dar uma visão geral e mais abstrata de como a aplicação vai funcionar e ser utilizada. Assim, a aplicação deverá permitir visualizar os locais a visitar bem como as informações associadas a este. Deverá também apresentar possíveis percursos que o utilizador possa gerar.



Figura 2 - Maquete

1.7. Definição de um conjunto de medidas de sucesso

A aplicação é um projeto no qual ambicionamos um crescimento não só a nível nacional como internacional.

Efetivamente, é esperado um crescimento exponencial na utilização da aplicação, não só pelo facto de a mesma se tornar mais conhecida e divulgada com o evoluir do tempo mas também com o esperado aumento de visitantes numa eventual situação pós pandémica.

No primeiro mês são esperados cerca de 700 downloads e acreditamos que alcançará milhares após o primeiro ano. Acreditamos também que cerca de 10% dos utilizadores se manterão ativos, uma vez que a restante percentagem poderá corresponder a cidadãos estrangeiros que após conhecerem a cidade não irão necessitar mais da aplicação.

1.8. Plano de desenvolvimento (diagrama GANTT)

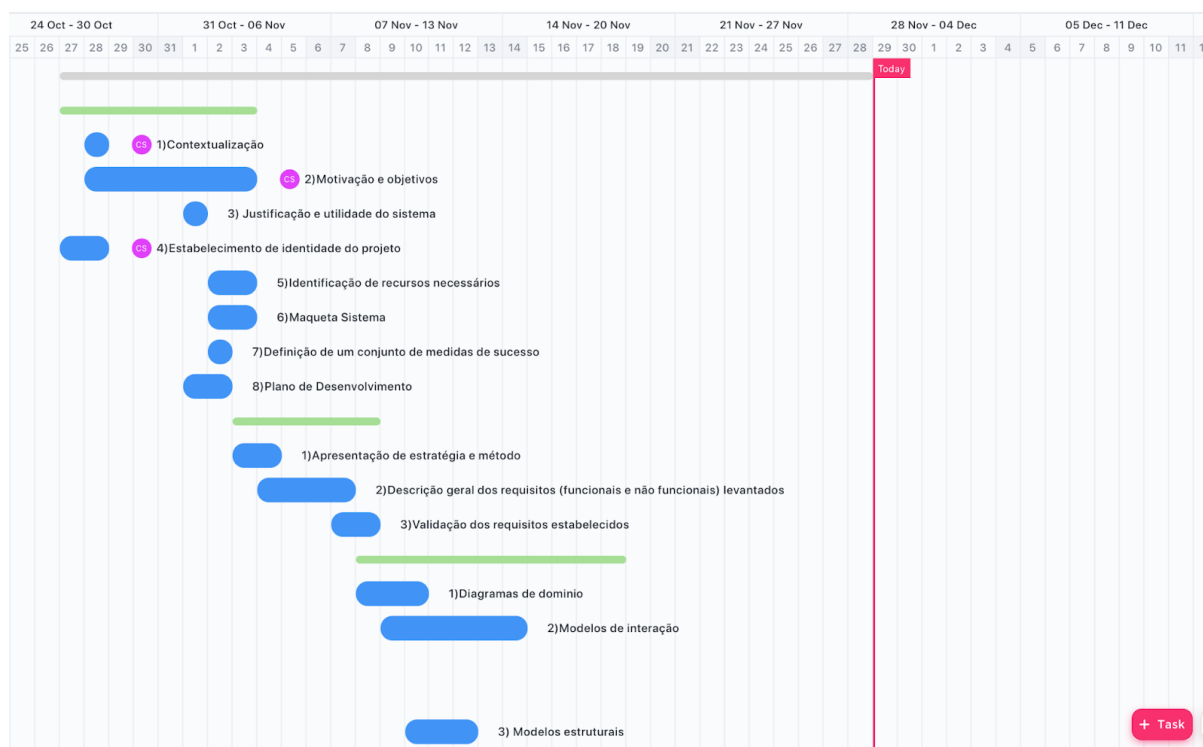
O projeto está dividido em duas partes distintas.

A especificação e fundamentação têm como objetivo analisar e projetar todos os requisitos e use cases operacionais e funcionais. Esta fase constitui uma parte fundamental para a próxima etapa através da criação de diagramas e modelos que vão servir de fundamentação para a implementação do software.

A construção foca-se no desenvolvimento do software que engloba a lógica de negócios, base de dados e a interface UI (construção). Concluída esta fase obtemos o produto final da aplicação pronto a ser usado pelo público-alvo.

Efetivamente, o diagrama de GANTT é uma ferramenta que ajuda a pôr em perspetiva o tempo que se tem disponível. Por isso, optámos por construir um diagrama de GANTT como forma de organizarmos o nosso projeto, respeitando as datas que nos foram impostas.

O primeiro diagrama (duas primeiras imagens) corresponde ao diagrama de GANTT da fase um. O segundo diagrama corresponde ao diagrama de GANTT da fase dois.



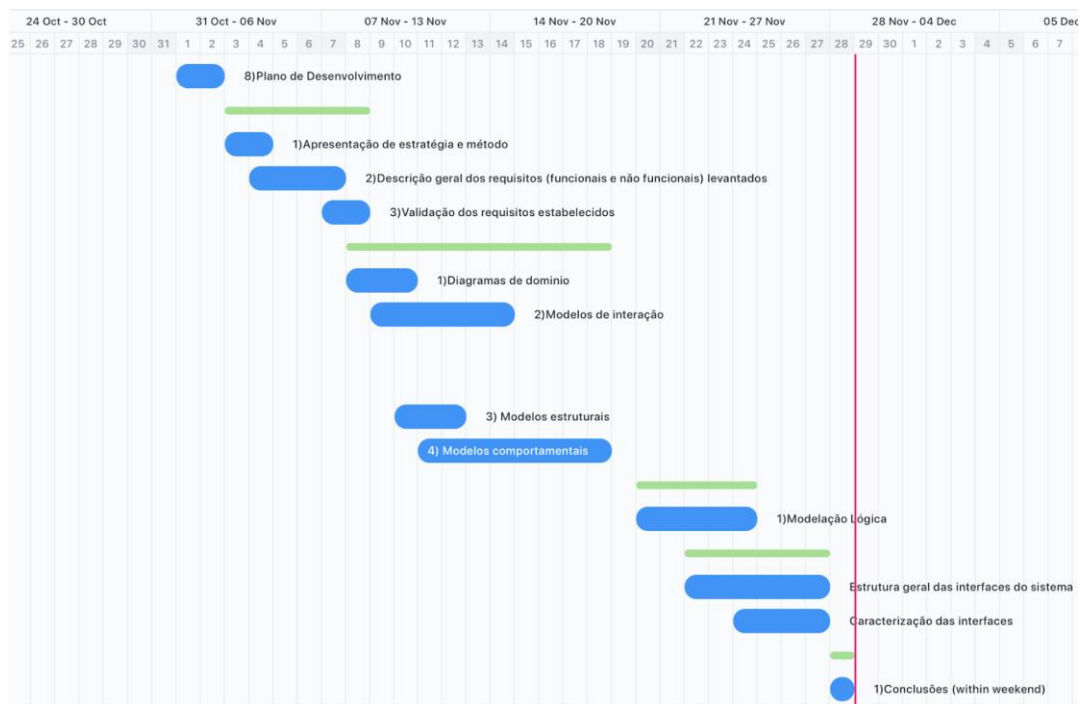


Figura 3 - Gantt parte1

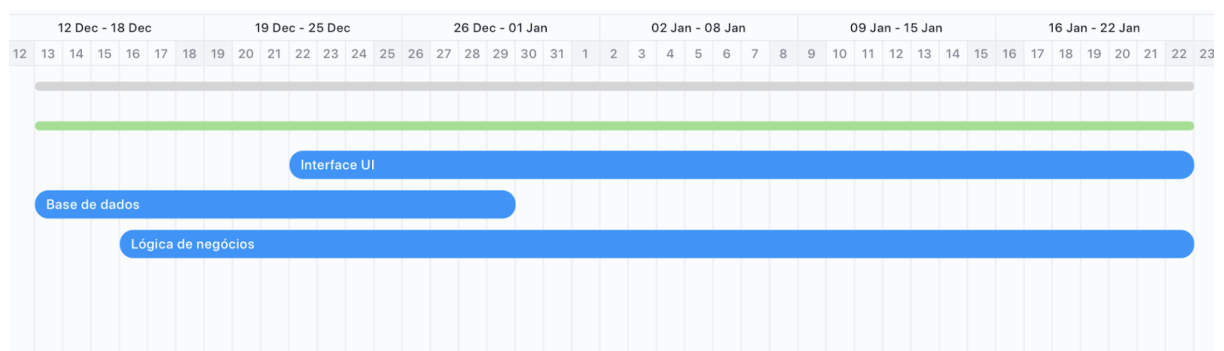


Figura 4 - Gantt parte 2

A fase um tem uma duração de aproximadamente um mês, enquanto que a fase dois tem duração de um mês e nove dias.

Relativamente à fase dois, consideramos que deve ser um trabalho diário constante, inicializando com a base de dados. Como boa prática, consideramos que se deveria reservar 2 dias no final da segunda fase para eventuais atrasos e contingências.

2. Levantamento e Análise de Requisitos

2.1. Apresentação da estratégia e método

Após definida a ideia geral do projeto que pretendemos desenvolver e de esboçar algumas das funcionalidades do programa, planeámos algumas reuniões com o presidente da câmara municipal de Braga cujas responsabilidades também englobam o pelouro da cultura e com os representantes dos diversos locais de interesse cultural (museus e teatros) com o objetivo de descobrir como estas atuavam até então e como publicitavam os locais que geriam. Adicionalmente, realizámos curtos inquéritos à comunidade, para fundamentar como este programa seria utilizado pela grande maioria dos utilizadores e o que estes gostariam de ver implementado. Com isto, desejávamos que o resultado deste projeto satisfizesse todas as entidades envolvidas.

Durante as reuniões realizadas com o presidente da câmara municipal e dos diversos locais de interesse cultural, foram solicitados diversos requisitos que o software a desenvolver deveria implementar. Um dos primeiros requisitos que as entidades culturais envolvidas exigiram foi a apresentação da informação dos locais a divulgar, mais concretamente criando um perfil do local que apresentasse a sua descrição, website, localização, horário de funcionamento e a sua avaliação de modo a que se torne mais acessível aos utilizadores conhecer e recolher informação dos locais, previamente. Foi-nos também solicitado, por parte do representante da câmara de Braga, que o software divulgasse também eventos temporários assim como festas populares de modo a que haja uma maior adesão.

Dos inquéritos concluímos que, de uma forma geral, quando estão a planear as suas viagens, os cidadãos, após se informarem um pouco sobre a região e sobre locais a visitar, procuram por percursos feitos por outros turistas e têm em consideração as opiniões e avaliações destes. Desta amostra da população, reparámos ainda que as pessoas se mostram mais interessadas em visitar certos locais quando nestes há eventos.

Assim, tendo em consideração os resultados deste inquérito, obtivemos uma lista de funcionalidades que seria interessante adicionar para atrair a atenção dos consumidores.

2.2. Descrição geral dos requisitos (funcionais e não funcionais) levantados

Requisitos Funcionais:

Com base nas reuniões e no inquérito realizado, obtivemos os seguintes requisitos a serem implementados:

2.2.1 Pesquisar locais a uma certa distância

Requisitos de Utilizador:

O utilizador indica o local, na cidade de Braga, em que se encontra ou o local que irá visitar e a aplicação indica os locais de interesse mais próximos. Estes devem ser restringidos por uma distância.

Requisitos de Sistema:

- *Sistema deve possuir a capacidade de obter a localização do utilizador;*
- *Sistema deve permitir que o utilizador introduza uma localização, na cidade de Braga, em formato de texto;*
- *Sistema deve permitir que o utilizador introduza uma distância máxima caso introduza simultaneamente a sua localização. Senão essa distância será determinada pelo zoom no mapa;*
- *Sistema deve apresentar os locais de interesse que se encontrem na área limitada por essa distância máxima/zoom.*

2.2.2 Pesquisar um local de interesse

Requisitos de Utilizador:

O utilizador indica o local de interesse (escrevendo o nome do local ou seleccionando no mapa) que pretende visitar e a aplicação deve mostrar as informações deste local

Requisitos de Sistema:

- Sistema deve permitir que o utilizador faça uma pesquisa por um local de interesse;
- Sistema deve dar opção ao utilizador de seleccionar o local de interesse no mapa;
- No sistema, cada local de interesse deve ter um nome único;
- Após seleccionar o local de interesse, o sistema deve apresentar o nome, a localização, uma descrição, horário de funcionamento, website, avaliação e eventos;
- Os locais de interesse têm gestores, contudo, estes não devem ser apresentados.

2.2.3 Sugerir percursos

Requisitos de Utilizador:

O utilizador solicita um percurso e o sistema gera-o com um certo número de locais de interesse indicado pelo utilizador. Este deve ser gerado em função de diferentes critérios como: distância, avaliação.

Requisitos de Sistema:

- Sistema deve pedir ao utilizador para indicar o número de locais que vão pertencer ao percurso;
- Sistema deve pedir ao utilizador para seleccionar um critério: escolher os locais de interesse mais próximos ou com melhores avaliações;
- Sistema deve mostrar ao utilizador o percurso total assinalado no mapa.

2.2.4 Avaliar locais de interesse

Requisitos de Utilizador:

O utilizador selecciona o local que pretende avaliar e atribui-lhe estrelas (de 1 a 5). O utilizador pode avaliar o mesmo local múltiplas vezes.

Requisitos de Sistema:

- Sistema deve permitir que o utilizador procure/selecione um local de interesse;
- O utilizador deve ser capaz de avaliar o local de interesse numa escala de 1 a 5 estrelas;
- Se o utilizador já tiver avaliado um local, o sistema deve substituir a avaliação já existente;
- A classificação média deve ser atualizada sempre que se alteram as classificações.

2.2.5 Funcionalidades do Gestor

Requisitos de Utilizador:

O gestor pode adicionar, editar e remover eventos do local a que está associado.

Requisitos do Sistema:

- Sistema permite ao gestor adicionar e remover eventos no local;
- Sistema permite ao gestor editar informação do local a que está associado e eventos;
- Sistema deve limitar estas operações ao gestor.

2.2.6 Funcionalidades do Timer

Requisitos do Sistema:

- Após a data de realização do evento, este deve ser dado como terminado e ser retirado do sistema.

Requisitos não Funcionais:

Obtivemos ainda alguns requisitos não funcionais, como: o programa deve funcionar 24h, deve ser capaz de obter a localização geográfica e o IP do utilizador. Deve também destacar os locais que apresentam eventos.

3. Especificação e Modelação do Software

Após concluído o levantamento e análise de requisitos e tendo sido feita a aprovação dos mesmos, demos início à modelação do software que pretendemos criar.

Neste capítulo, iremos desenvolver alguns diagramas UML que irão ser uma representação abstrata das funcionalidades que retiramos dos requisitos levantados no capítulo anterior.

Decidimos apresentar 4 tipos de diagramas: diagramas de contexto, diagramas de interação, diagramas estruturais e diagramas comportamentais.

3.1. Diagramas de contexto

Começamos a especificação e modelação do software pelo desenvolvimento de diagramas de contexto. Estes diagramas permitem-nos visualizar e limitar a área em que queremos trabalhar e, conhecendo as funcionalidades que desejamos implementar, consideramos que seria um passo importante a tomar antes de começar a analisar em mais detalhe a estrutura da nossa aplicação.

3.1.1 Modelo de domínios

Dentro dos diagramas de contexto, optámos por desenvolver modelos de domínios, dado que estes apresentam de uma forma muito abstrata os objetos que consideramos essenciais para a implementação do software.

Assim, criámos o modelo de domínios que melhor representa o nosso projeto, a área em que estamos a trabalhar e as entidades que pretendemos representar.

Com base nos requisitos, concluímos que necessitávamos de definir uma entidade geral que representasse museus e teatros, denominada Local. Um Local tem associado um nome, localização, descrição, horário, website, classificação e pode conter, ou não, eventos que, por sua vez, têm atribuídos um nome, descrição e a data e hora em que se vão realizar.

Cada local é gerido por um gestor, que para alterar os dados do local e definir novos eventos irá precisar de fazer uma autenticação. Assim, cada gestor tem associado duas entidades, E-mail e Password.

Por último, considerámos também que iríamos precisar de uma representação da entidade Percurso. Esta iria assimilar-se a uma lista de Local.

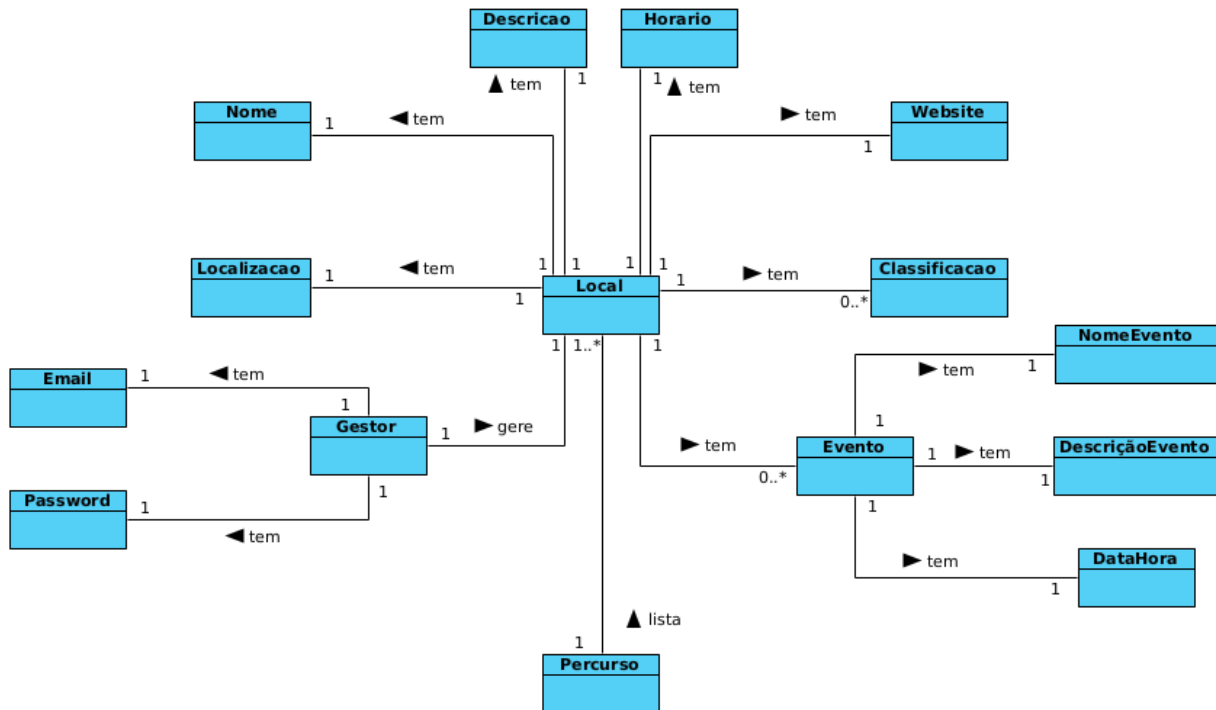


Figura 5 - Modelo de Domínios

3.2. Modelos de interação

Na sequência dos modelos de contexto, iremos agora explorar os modelos de interação.

Estes modelos permitem representar como o utilizador poderá interagir com o nosso software e como os componentes do nosso sistema irão interagir entre eles.

Modelos de interação incluem diagramas de use case e de sequência.

3.2.1 Diagrama de Use Cases

Os diagramas de use case permitem-nos definir como o utilizador poderá interagir com o software e, consequentemente, como o sistema deverá reagir aos pedidos do utilizador.

Estes diagramas são uma representação estruturada dos requisitos que levantamos e permitiram-nos analisar com mais detalhe que funcionalidades devemos implementar.

1. Use Case : Pesquisar locais

Pré-condição: True

Pós condição: Sistema disponibiliza vários locais a uma certa distância da localização indicada

Fluxo Normal:

1. Sistema pede localização do utilizador;
2. Utilizador permite ao sistema aceder à sua localização;
2. Utilizador indica a distância máxima a que se podem apresentar os locais, fazendo zoom;
3. Sistema verifica que a localização dada é em Braga;
4. Sistema faz procura dos locais na área definida pelo utilizador;
5. Sistema apresenta os vários locais de interesse nessa região;

Fluxo Alternativo 1 [Sistema não consegue aceder à localização] (passo 2):

- 2.1. Sistema avisa utilizador que não conseguiu aceder à sua localização;
- 2.2. Sistema pede a utilizador para introduzir a sua localização manualmente e o raio pretendido;
- 2.3. Voltar para passo 4 do fluxo normal;

Fluxo Alternativo 2 [Localização do utilizador não é em Braga] (passo 4):

- 4.1 Sistema informa que utilizador não se encontra em Braga;
- 4.2 Sistema atualiza localização para o centro de Braga;
- 4.3 Voltar para o passo 5 do fluxo normal;

Fluxo de Exceção 3 [Sistema não encontra locais no raio definido] (passo 6):

- 6.1. Sistema avisa utilizador de que no raio definido não existem locais de interesse;

2. Use Case: Pedir um Percurso

Pré-condição: True

Pós-condição: Sistema gera um percurso

Fluxo Normal:

1. Sistema solicita o número de locais que o utilizador pretende visitar;
2. Sistema pede ao utilizador para indicar o critério de escolha dos locais;
3. Utilizador indica ao sistema o número de locais e o critério de escolha;
4. Sistema gera e apresenta um percurso de acordo com os critérios indicadas pelo utilizador;

Fluxo Alternativo 1 [Utilizador indica um número maior que os pontos de interesse existentes] (passo 3):

- 3.1. Sistema indica que apenas é possível gerar um percurso com o maior número de pontos de interesse existentes;
- 3.2. Sistema pergunta ao utilizador se pretende gerar o percurso com todos os pontos de interesse existentes;
- 3.3. Utilizador aceita sugestão do sistema;
- 3.4. Volta ao passo 4;

Fluxo de Exceção 2 [Utilizador indica ao sistema que não pretende gerar um percurso com todos os pontos de interesse existentes] (passo 3.3):

3.3.1. Sistema não gera um percurso;

3. Use Case: Avaliar local

Pré-condição: True

Pós-condição: Avaliações do local atualizadas

Fluxo Normal:

1. Utilizador indica o local que pretende avaliar;
2. Sistema pergunta quantas estrelas pretende atribuir ao local;
3. Utilizador indica um valor entre 1 a 5 estrelas;
4. Sistema obtém o IP do utilizador para associar à classificação;
5. Sistema atualiza as classificações do local e calcula a nova média de classificações;

Fluxo de Exceção 1 [Local de interesse não existe] (passo 1):

- 1.1. Utilizador não avalia local;

Fluxo de Alternativo 2 [Utilizador não atribui um valor entre 1 e 5] (passo 3):

- 3.1. Sistema informa que o valor não é válido;
- 3.2. Volta ao passo 2.

4. Use Case: Consultar local de interesse

Pré-condição: True

Pós-condição: Um local de interesse foi consultado

Fluxo Normal:

1. Utilizador seleciona no mapa ou indica o nome do local de interesse que pretende consultar;
2. Sistema apresenta a informação disponível sobre o local indicado (nome, localização, descrição, horário de funcionamento, website, avaliação e eventos);

Fluxo de Exceção 1 [Local de interesse não existe na base de dados] (passo 1):

- 1.1. Sistema informa o utilizador que o nome do local de interesse indicado não existe na base de dados;

5. Use Case: Login Gestor

Pré-condição: True

Pós condição: Gestor está autenticado;

Fluxo Normal:

1. Sistema pede o e-mail e a password;
2. Sistema valida os dados;
3. Gestor está autenticado.

Fluxo de Exceção 1 [Dados não são válidos] (passo 2):

- 2.1. Sistema informa que dados não são válidos;

6. Use Case: Adicionar evento

Pré-condição: Gestor autenticado

Pós-condição: É adicionado um evento à lista de eventos de um determinado local

Fluxo Normal:

1. Gestor indica o nome do novo evento que pretende criar;
2. Sistema pede mais informações sobre o evento;
3. Gestor indica data, hora e descrição;
4. Sistema adiciona evento à lista de eventos no local que este gere;

Fluxo de Exceção 1 [Já existe um evento com o nome dado] (passo 1):

- 1.1. Sistema informa que já existe um evento com esse nome;
- 1.2. Sistema não cria um novo evento.

Fluxo de Exceção 2 [Data e Hora inválida] (passo 3):

- 3.1. Sistema informa que não é possível definir um evento na data e hora indicada;
- 3.2. Sistema não cria um novo evento.

7. Use Case: Remover evento

Pré-condição: Gestor autenticado

Pós-condição: É retirado um evento à lista de eventos de um determinado local

Fluxo Normal:

1. Gestor indica o nome do evento que pretende remover;
2. Sistema remove evento.

Fluxo de Exceção 1 [Não existe nenhum evento com o nome inserido] (passo 1):

- 1.1. Sistema informa que não existe nenhum evento com o nome inserido;
- 1.2. Sistema não remove evento.

8. Use Case: Fim do evento

Pré-condição: Data do evento já passou

Pós-condição: É retirado um evento à lista de eventos de um determinado local de interesse

Fluxo Normal:

1. Timer notifica o Sistema do fim do evento;
2. Sistema retira evento.

9. Use Case: Editar Informações de um Local

Pré-condição: Gestor autenticado

Pós-condição: Informações do local a que o Gestor está associado são atualizadas

Fluxo Normal:

1. Sistema apresenta informações do local;
2. Gestor informa que campo pretende alterar;
3. Gestor adiciona a nova informação;
4. Informação do local atualizada.

10. Use Case: Editar Informações de um Evento

Pré-condição: Gestor autenticado

Pós-condição: Informações de um evento do local a que o Gestor está associado atualizadas

Fluxo Normal:

1. Sistema pergunta ao gestor que evento pretende editar;
2. Gestor indica o nome do evento que pretende editar;
3. Sistema apresenta informações do evento selecionado;
4. Gestor informa que campo pretende alterar;
5. Gestor adiciona a nova informação;
6. Informação do evento atualizada.

Com os Use Cases definidos, desenvolvemos o seguinte diagrama de Use Cases:

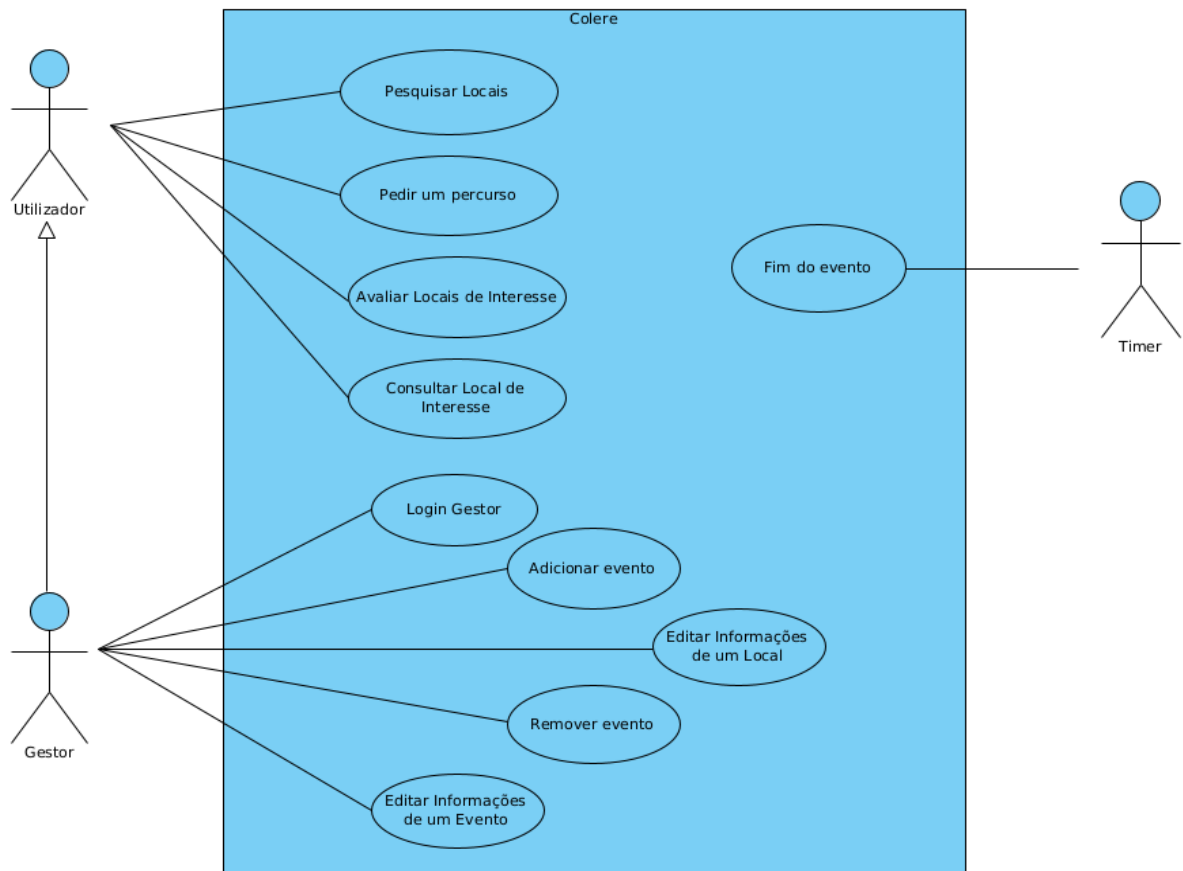


Figura 6 - Diagrama de Use Cases

3.2.2 Diagramas de Sequência

Para completar os modelos de interação, desenvolvemos os diagramas de sequência.

Optamos por definir estes diagramas dado que são uma ótima forma de complementar o diagrama de use cases, uma vez que este representa como o utilizador interage com o sistema e os diagramas de sequência além de permitirem identificar como os atores interagem com o sistema permitem observar como os diferentes objetos interagem entre si.

Antes de dar início ao desenvolvimento dos diagramas de classes, definimos algumas entidades do software com base no modelo de domínios e acrescentamos uma entidade “ColereFacade” que se irá comportar como uma entidade principal. Iremos recorrer a esta para executar as principais funcionalidades e esta conhecerá alguns dos dados essenciais para a execução das funcionalidades. Esta entidade será explicada com mais detalhe no subcapítulo 3.3.1.

1. Pesquisar Locais:

Para implementar a funcionalidade que permite a um utilizador pesquisar locais, os quais se devem encontrar em Braga, precisamos de obter a sua localização, seja acendendo à localização atual do utilizador ou permitindo que este a escreva.

Neste diagrama de sequências, assume-se que as condições do Use Case associado são verificadas.

De seguida, devemos percorrer todos os locais definidos no sistema e, para cada um deles, obter a sua localização. Se a distância entre as duas localizações for menor ou igual à indicada, o local deve estar presente na resposta dada ao utilizador. É importante relembrar que a lista de locais deve estar ordenada pela sua distância ao utilizador e que os locais que tenham eventos devem ser destacados no UI.

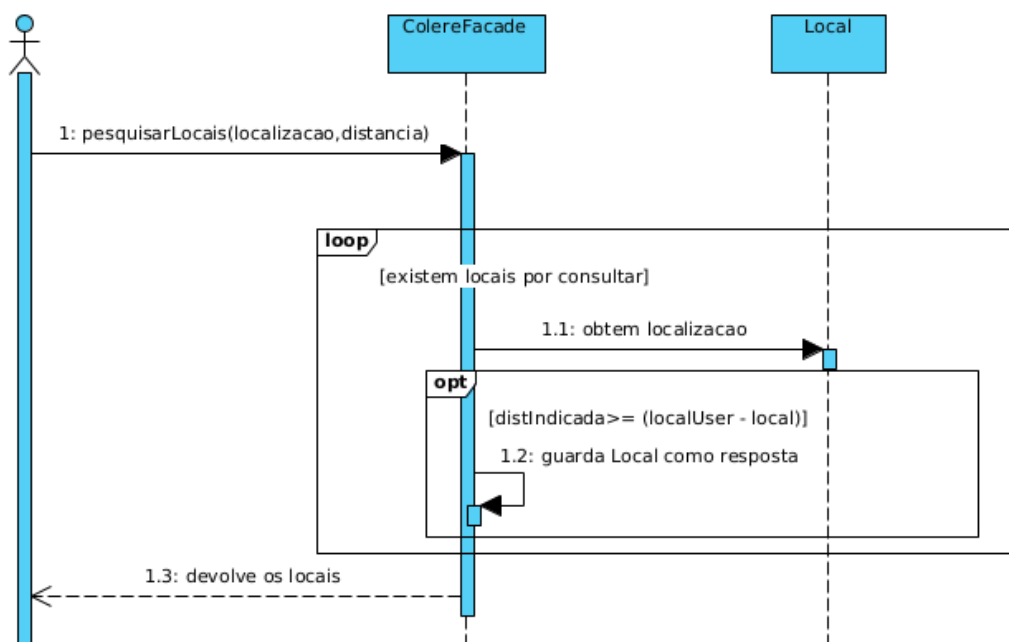


Figura 7 - Diagrama de sequências 1

2. Pedir um Percurso:

Quando um utilizador pede um percurso, ele deve indicar quantos locais deseja no percurso e segundo que critério o percurso deve ser gerado.

Com estes argumentos, iremos consultar todos os locais existentes no sistema. Enquanto tivermos menos locais no percurso que a quantidade de locais indicada, iremos adicionar todos os locais ao percurso. Quando o percurso tiver o número suficiente de locais, teremos de escolher os melhores locais segundo o critério indicado.

Portanto, para cada local, iremos consultar o valor (distância ou classificação) associado a esse. Se esse valor for melhor que o pior valor de um local presente no percurso, o local em questão deverá ser substituído.

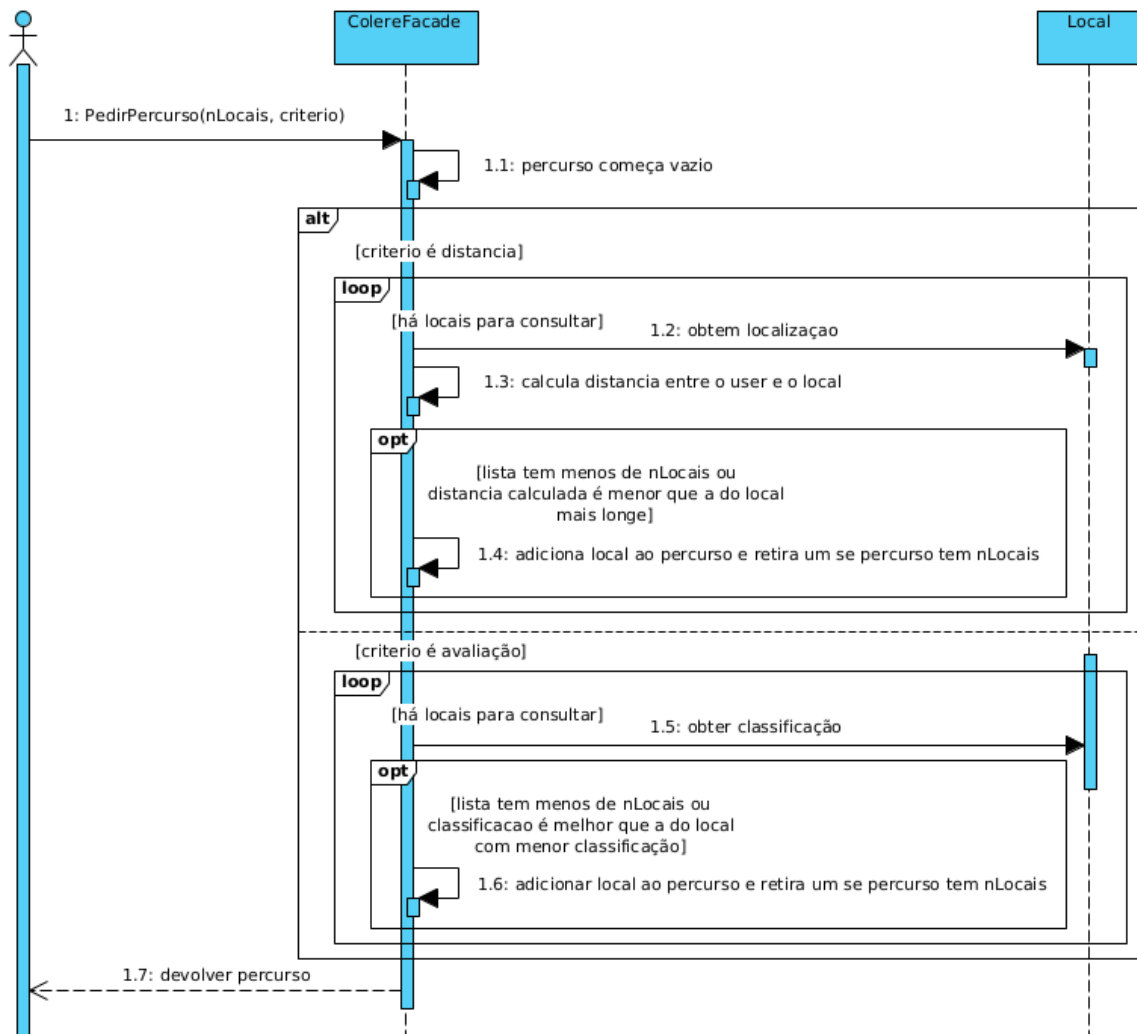


Figura 8 - Diagrama de sequências 2

3. Avaliar local de interesse:

Para que o utilizador possa avaliar um local, este deve indicar o local que pretende avaliar e uma classificação de 1 a 5. O sistema deve obter o local em questão e adicionar na sua classificação um novo valor, tal como recalcular a classificação média atual. A avaliação do utilizador ficará associada ao seu IP. Será ainda possível que um utilizador reavalie um dado local, logo, caso o IP já exista na classificação do local, o valor de 1 a 5 que lá se encontra deve ser substituído.

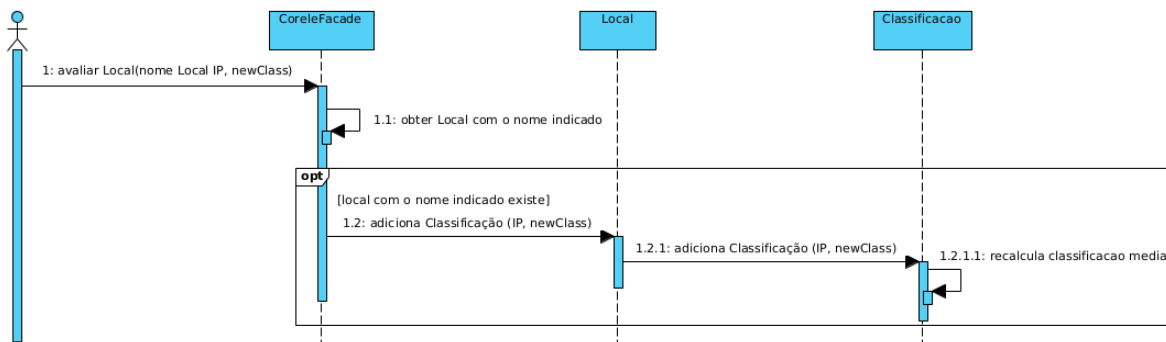


Figura 9 - Diagrama de sequências 3

4. Consultar Local de Interesse:

Quando um utilizador pretender visualizar a informação detalhada de um local, deve indicar o nome deste ou seleccionar no mapa. Caso opte por seleccionar no mapa, como este local está associado ao seu nome, pode atualizar as mesmas funções que a procura pelo nome.

Deverá ser feito uma procura na base de dados do sistema por um local com o nome indicado.

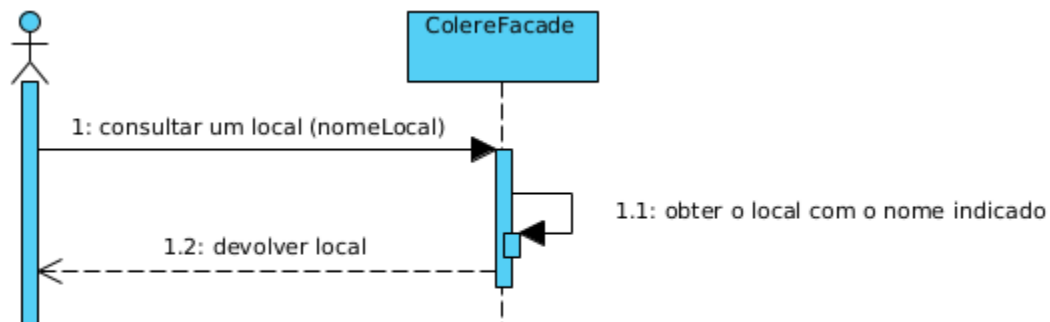


Figura 10 - Diagrama de sequências 4

5. Login Gestor:

O gestor é a entidade responsável por um local de interesse, portanto tem funcionalidades que lhe são exclusivas. Para poder aceder a essas funcionalidades, tem que fazer login. Para fazer este login, o ator deve indicar o seu e-mail e a sua password.

Comparando os dados introduzidos com os dados do sistema, podemos autenticar o gestor.

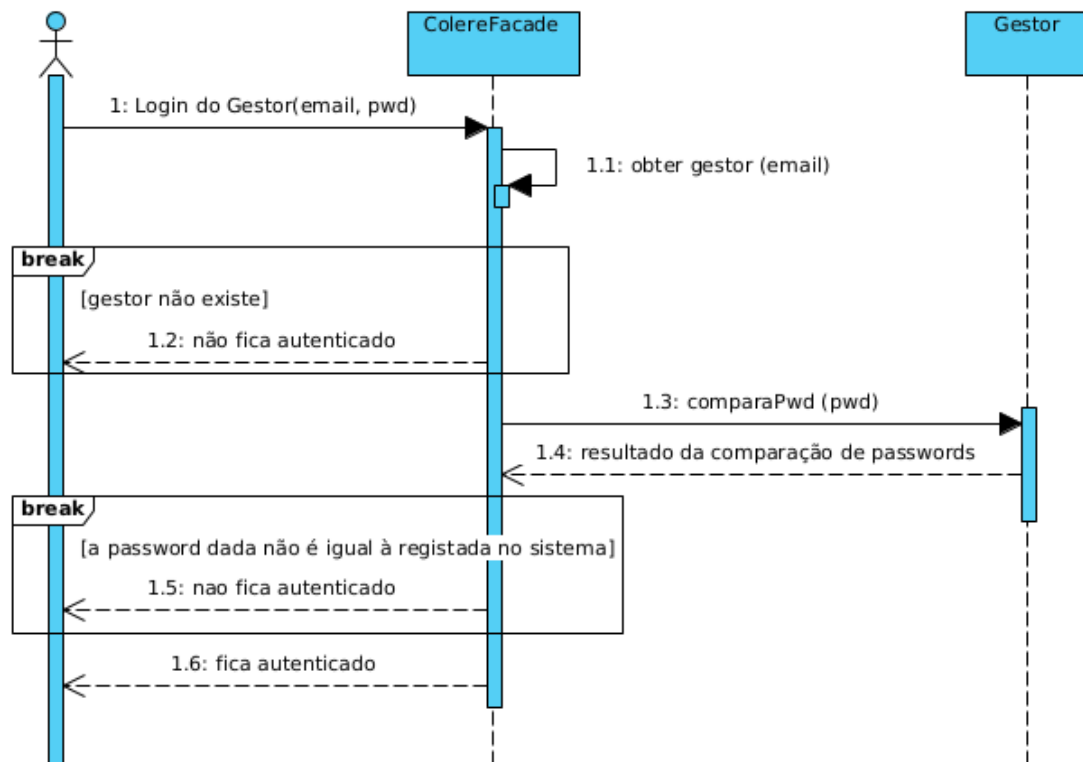


Figura 11 - Diagrama de sequências 5

6 e 7. Adicionar e Remover Evento:

Quando um gestor pretender adicionar um novo evento, este deve fazê-lo no local que gere, isto é, não pode alterar eventos de locais sobre os quais não tem autoridade. Assim, como o objeto gestor conhece o local com o qual pode trabalhar, este acede-lhe e cria um novo evento que posteriormente será adicionado à lista dos eventos deste local. Para tal, o gestor deve indicar o nome, a data e descrição e este novo evento deve ser criado no local em questão.

A operação de remover evento será muito semelhante à de adicionar, dado que, uma vez mais, o gestor só pode alterar os dados do local que gere e como conhece o local para o qual trabalha, facilmente poderá remover um evento da base de dados do mesmo, indicando o nome do evento.

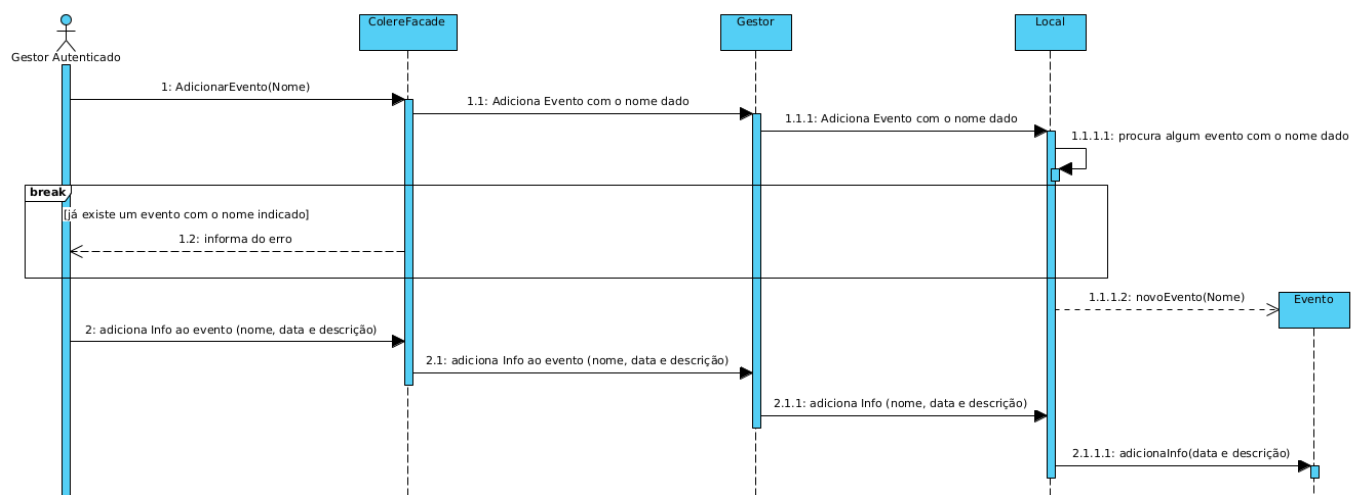


Figura 12 - Diagrama de seqüências 6

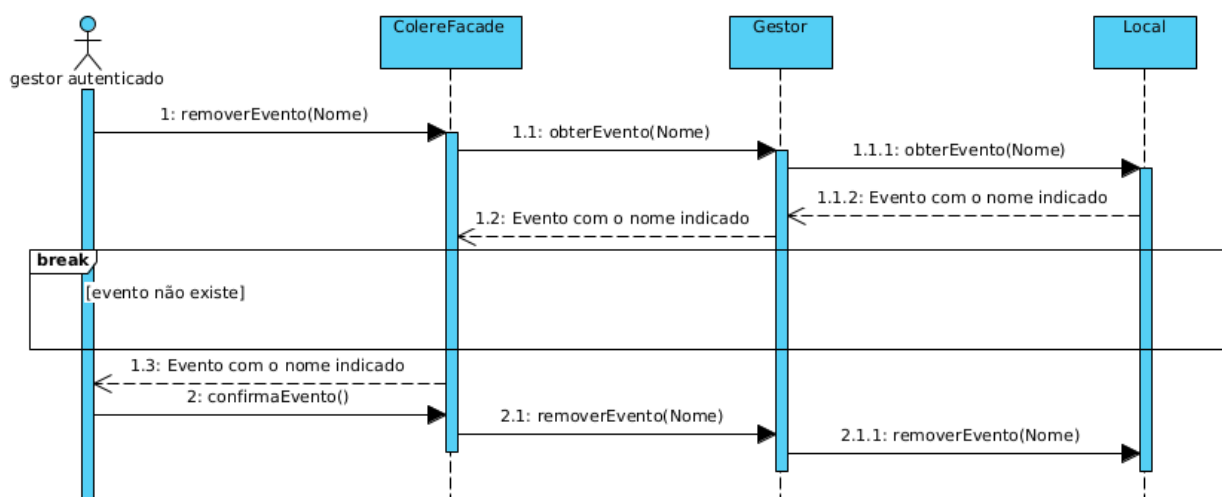


Figura 13 - Diagrama de seqüências 7

8. Use Case: Fim do evento

Pretendemos que, quando um evento chega ao fim, o sistema o remova. Assim, devem ser consultados todos os locais e todos os eventos destes e caso já tenham ocorrido, devem ser removidos.

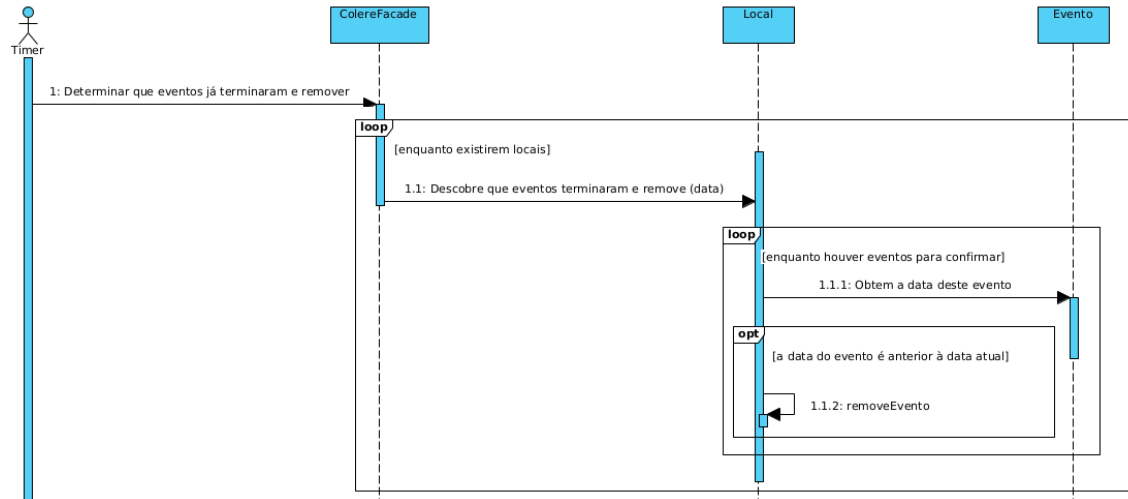


Figura 14 - Diagrama de sequências 8

9. Use Case: Editar Informações de um Local

Como o gestor está autenticado, podemos aceder às informações do local que ele gere. Após apresentarmos estes dados ao Gestor autenticado, ele pode indicar que dados pretende alterar e qual a informação nova que pretende que seja adicionada. Esta informação será adicionada à base de dados.

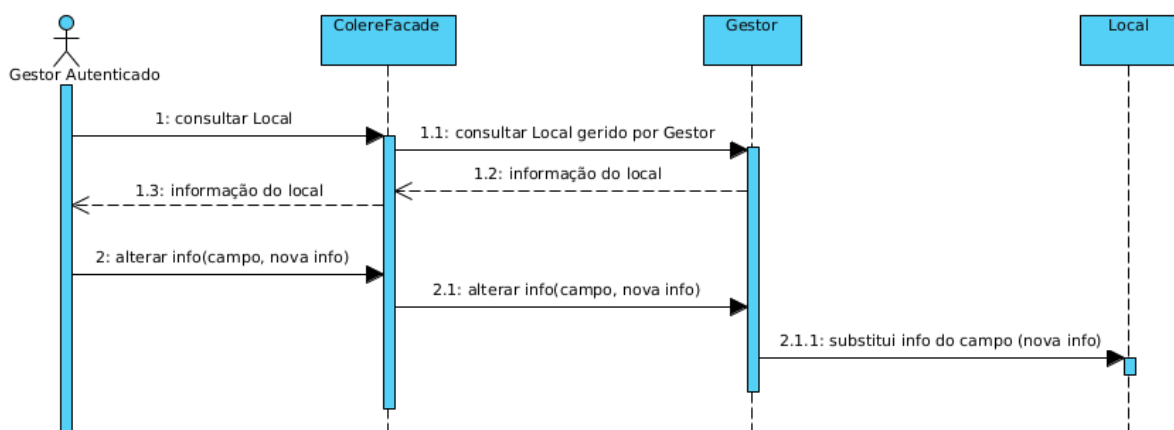


Figura 15 - Diagrama de sequências 9

10. Use Case: Editar Informações de um Evento

À semelhança do use case anterior, o gestor também pode consultar a informação dos eventos que irão ocorrer no local que gere. Inicialmente, serão-lhe apresentados todos os eventos que estão planeados e após indicar o nome do evento que pretende alterar, serão-lhe apresentadas todas as informações do mesmo. O gestor deverá indicar que critério pretende alterar e a nova informação que deve ser adicionada ao sistema. O evento deverá ser devidamente atualizado.

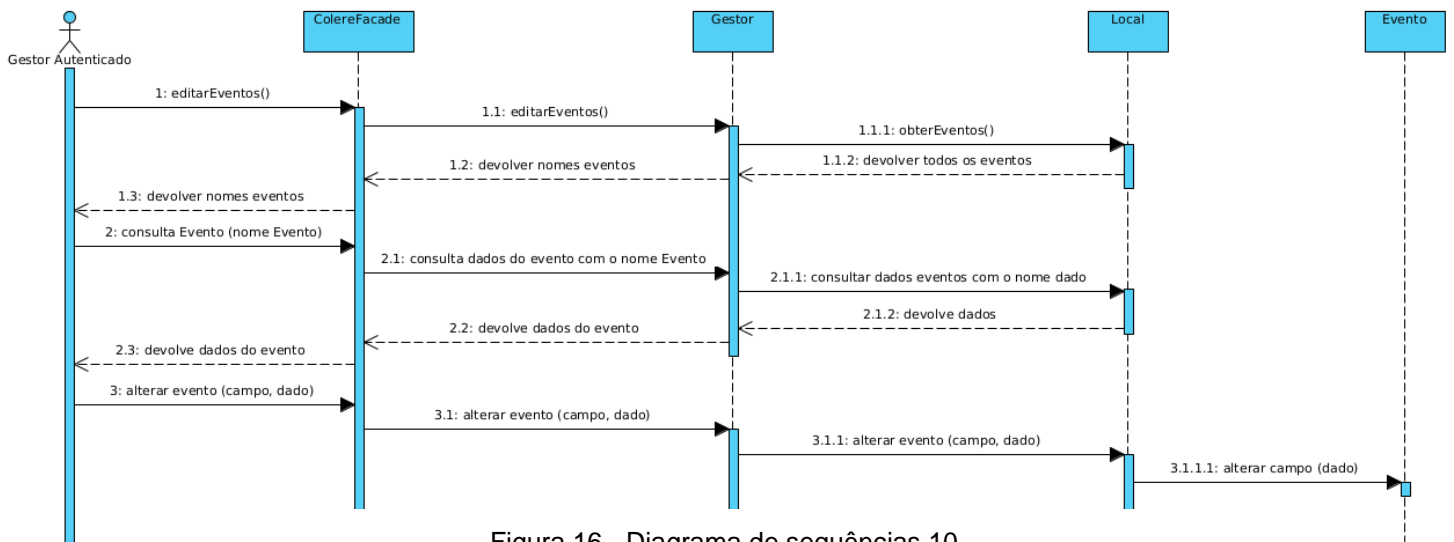


Figura 16 - Diagrama de sequências 10

3.3. Modelos Estruturais

Os modelos estruturais permitem apresentar a organização do sistema e como este é constituído, isto é, quais os seus componentes e as relações entre eles. Permitem-nos ter uma melhor perceção dos métodos a implementar de modo a obter a aplicação final.

3.3.1 Diagrama de Classes

Optamos por desenvolver um diagrama de classes porque este nos permite visualizar em detalhe como estas irão interagir entre elas, assim como, definir algumas das funções que consideramos necessárias e que foram resultados da análise dos diagramas dos subcapítulos anteriores.

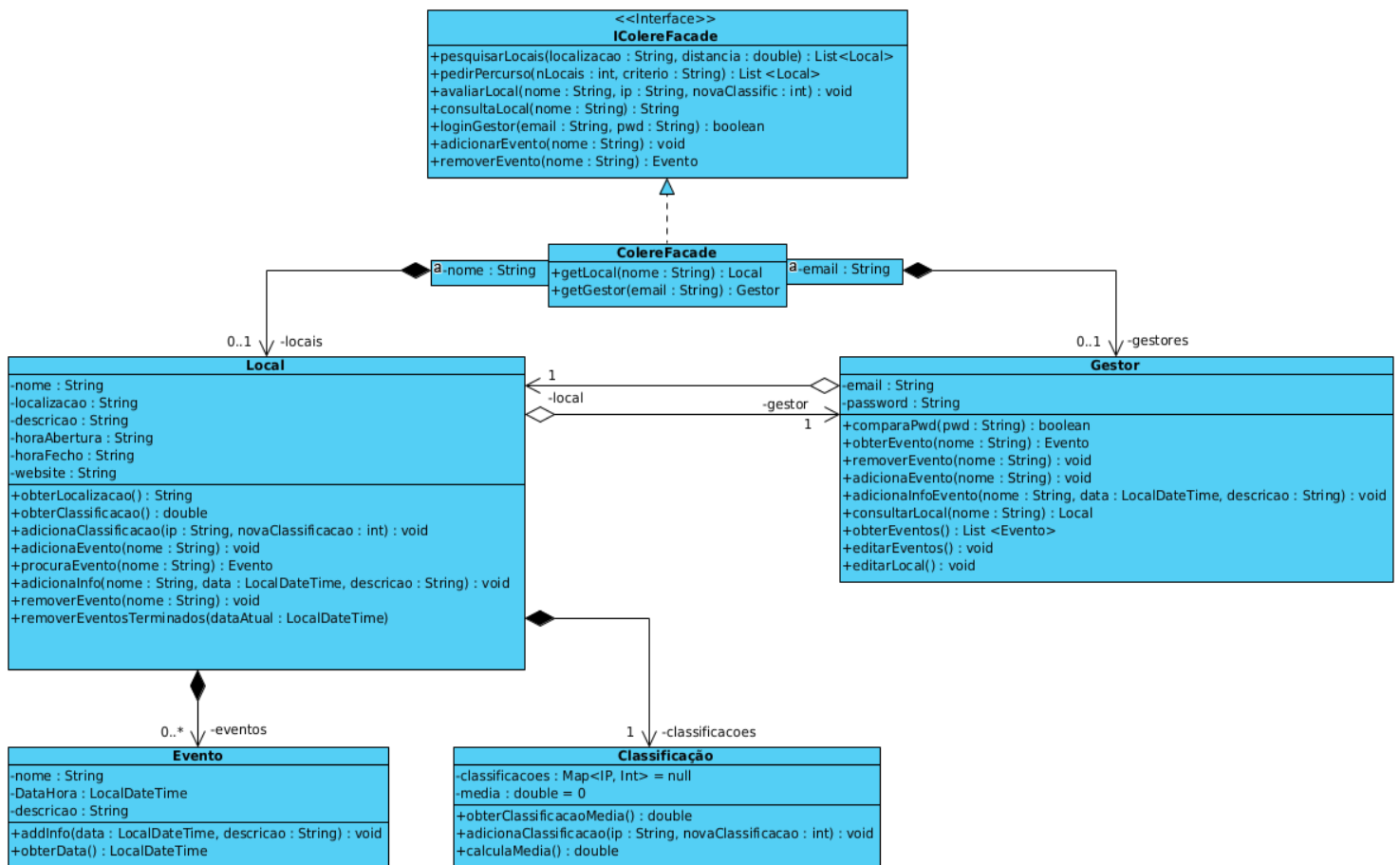


Figura 17 - Diagrama de Classes

3.4. Modelos Comportamentais

Na sequência dos diagramas de interação, desenvolvemos os diagramas comportamentais que permitem projetar o software.

De facto, os diagramas comportamentais são a representação de um estado ou situação em que um objeto se encontra no decorrer da execução dos processos de um sistema. É uma maneira eficiente e clara de descrever todos os possíveis estados de um sistema, assim como que eventos levam à transição de um estado para outro.

3.4.1 Diagramas de Atividades

Na realidade, optámos por usar diagramas de atividades, uma vez que são os que achamos mais adequados. Diagramas de atividades permitem descrever um processo, através do encadeamento das ações que o compõem.

1. Pesquisar Locais

O sistema acede à localização do utilizador sendo que a distância máxima a que se podem apresentar os locais depende do zoom que é feito no mapa. No entanto, caso o sistema não consiga aceder à localização do utilizador, o mesmo deve ser avisado e deve-lhe ser pedido para inserir a localização manualmente. De seguida, o sistema verifica se a localização é em Braga e se não se verificar, deve atualizar a localização para o centro da cidade. Por fim, o sistema apresenta os vários locais de interesse nessa região. No entanto, se o sistema não encontrar locais no raio definido o utilizador é informado.

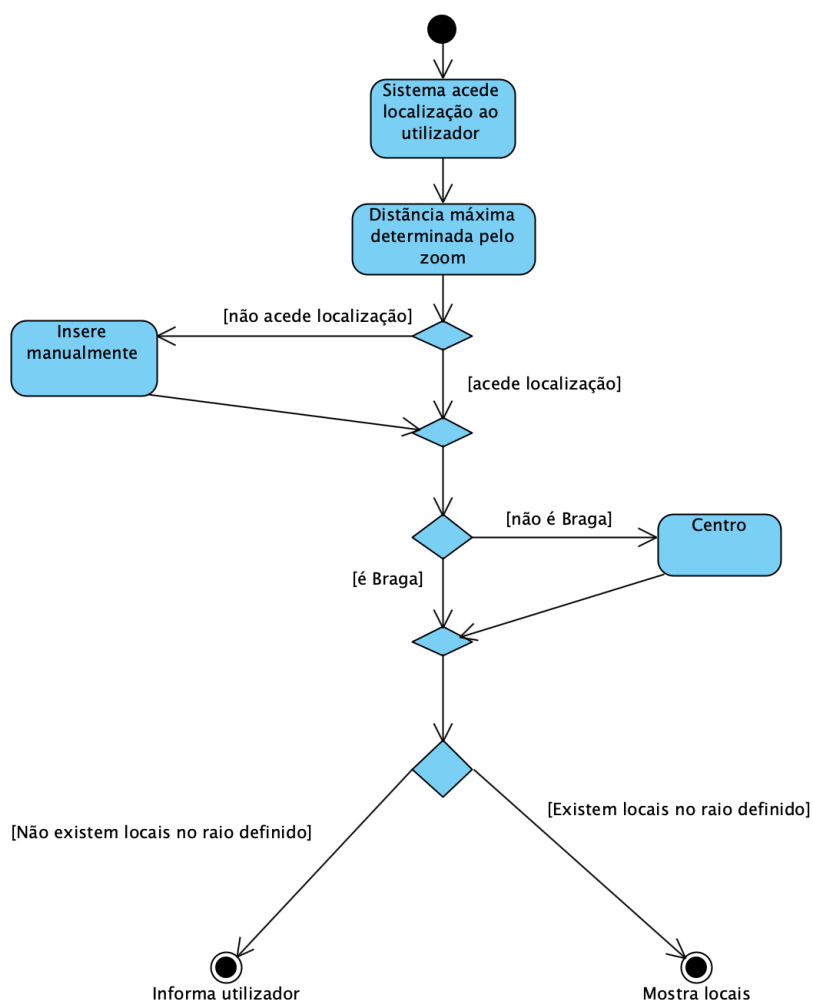


Figura 18 - Diagrama de atividades 1

2. Avaliar Local

Para avaliar o local na aplicação que idealizamos, o utilizador indica o nome do local e as estrelas com que o pretende avaliar (1-5). Após atribuir as estrelas, o sistema descobre o ip e, posteriormente, adiciona a classificação e atualiza as avaliações.

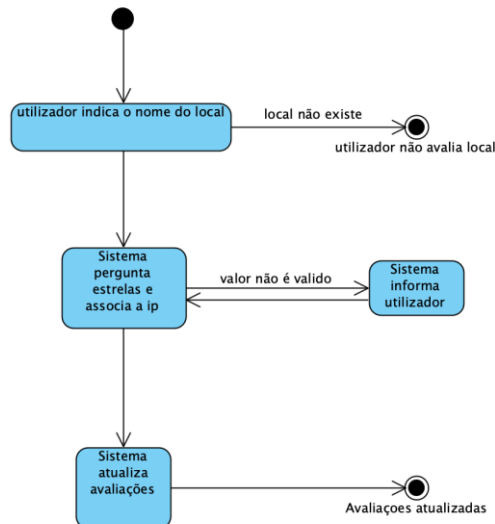


Figura 19 - Diagrama de atividades 2

3. Pedir percurso

O sistema começa por pedir ao utilizador a quantidade de pontos de interesse que o mesmo pretende. Se não existir a quantidade de pontos que solicitou, informa-se o utilizador da quantidade de pontos existentes. Posteriormente, pode escolher entre gerar um percurso com o máximo de pontos existentes ou não gerar percurso.

Para continuar a gerar o percurso, o sistema pede ao utilizador para indicar o critério de escolha dos pontos. Por fim, o sistema gera o percurso.

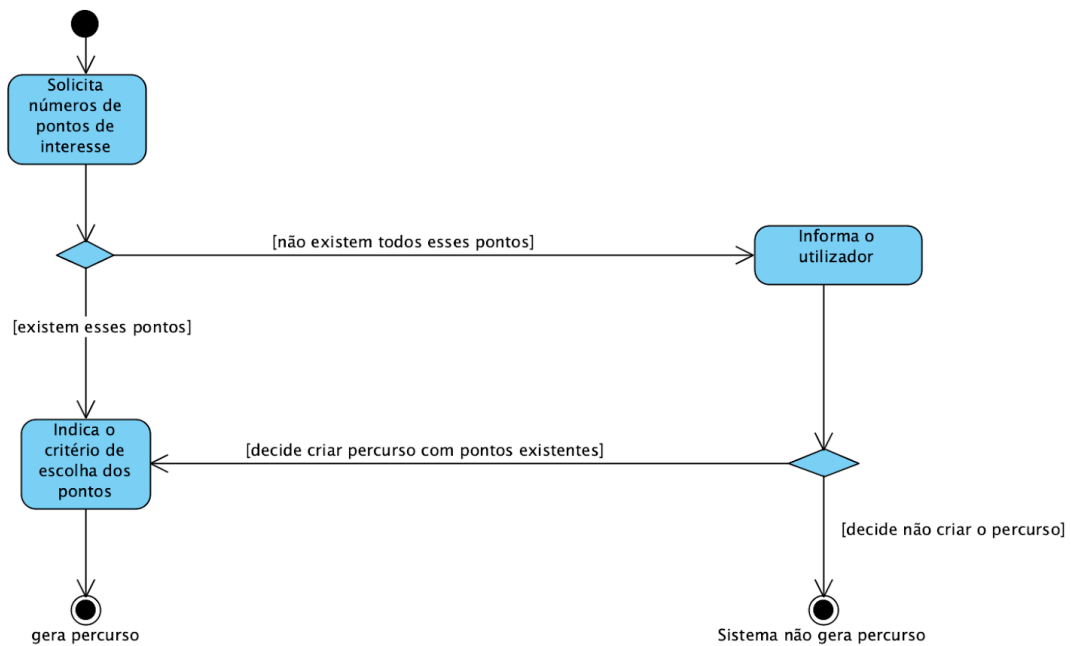


Figura 20 - Diagrama de atividades 3

4. Consultar locais

O utilizador começa por indicar o local que pretende consultar, podendo fazê-lo carregando no mapa o lugar ou indicando o nome do local. Caso esse local não exista, o sistema informa o utilizador que não existe esse lugar, se não, a informação sobre o local é apresentada.

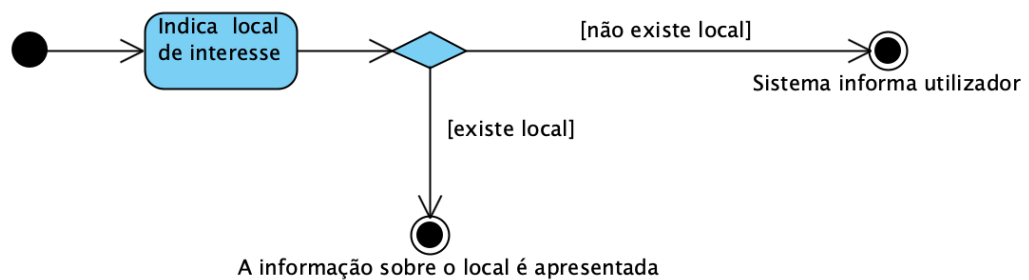


Figura 21 - Diagrama de atividades 4

5. Login Gestor

Inicialmente o gestor indica que pretende fazer login. O sistema pergunta o email e a password. Se os dados forem válidos o gestor fica autenticado, caso contrário o sistema informa que dados são inválidos.

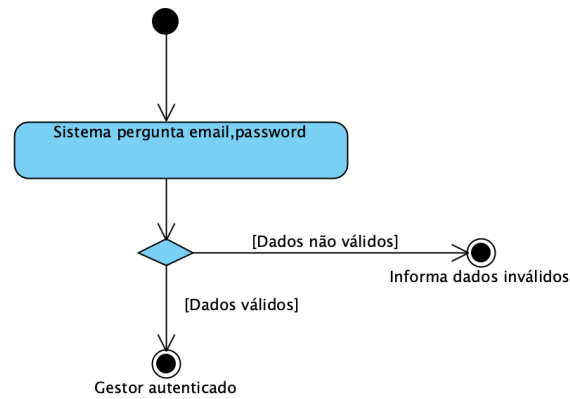


Figura 22 - Diagrama de atividades 5

6. Adicionar evento

O gestor é a única entidade que pode adicionar um evento. Para tal, o sistema pergunta o nome, a data e hora e a descrição do evento. Se a data ou nome inseridos forem inválidos, o sistema não adiciona nenhum evento.

Por fim, o evento é adicionado.



Figura 23 - Diagrama de atividades 6

7. Remover evento

O evento pode ser retirado pelo próprio gestor ou pelo sistema. Se for retirado pelo gestor, o mesmo pede para remover um evento, indicando o nome deste, e o mesmo é retirado. Por outro lado, o sistema pode remover o evento caso já tenha passado a data do mesmo.

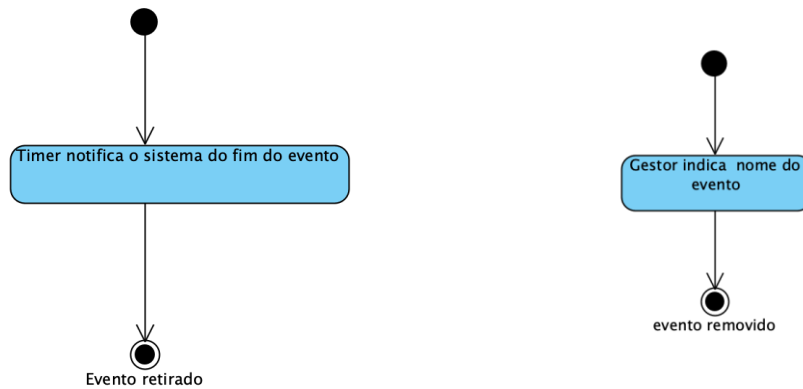


Figura 24 - Diagramas de atividades 7

8. Editar informações de um local

Após o gestor informar que pretende editar as informações do local, o sistema apresenta-as ao Gestor permitindo editá-las. Por fim, o Gestor edita as informações e as mesmas são atualizadas.

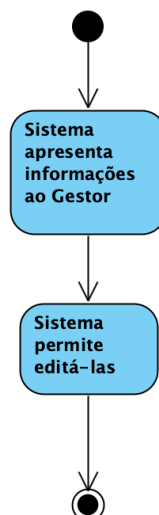


Figura 25 - Diagrama de atividades 8

9. Editar informações de um Evento

O gestor é a única entidade que pode editar um evento. Desta maneira o sistema pergunta ao gestor que evento pretende alterar. Após ter escolhido o evento, o sistema apresenta todas as informações sobre o mesmo e permite ao gestor alterar. Após editar o que for preciso as informações são atualizadas.

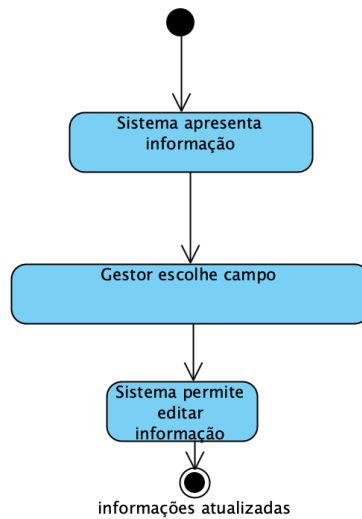


Figura 26 - Diagrama de atividades 9

4. Conceção do Sistema de Dados

Uma base de dados é uma ferramenta que serve para armazenar informação. Esta informação relaciona-se segundo certas regras e possui uma determinada finalidade.

Posto isto, com a implementação da nossa base de dados, pretendemos armazenar toda a informação relevante ao nosso problema de modo a garantir um melhor funcionamento da nossa aplicação.

Modelação lógica

- **Local**

Um local possui obrigatoriamente um nome, o seu horário de funcionamento, a sua localização e a sua respetiva avaliação média. Pode ainda conter informação sobre o seu website e uma breve descrição. Possui também, obrigatoriamente, o gestor a esse local associado.

Cada local, pode ou não conter eventos. O mesmo se aplica às classificações atribuídas pelos utilizadores.

- **Evento**

Um evento, possui obrigatoriamente um nome, a data e a hora a que se vai realizar e o local a que este está associado, ou seja, onde vai decorrer. Pode também possuir uma breve descrição do mesmo.

- **Gestor**

Cada gestor gere um e apenas um local e possui obrigatoriamente um email e uma password de modo a que seja possível realizar a sua autenticação.

- **Classificação**

Uma classificação possui obrigatoriamente uma string que contém o IP, um valor de 1 a 5 e o Local avaliado.

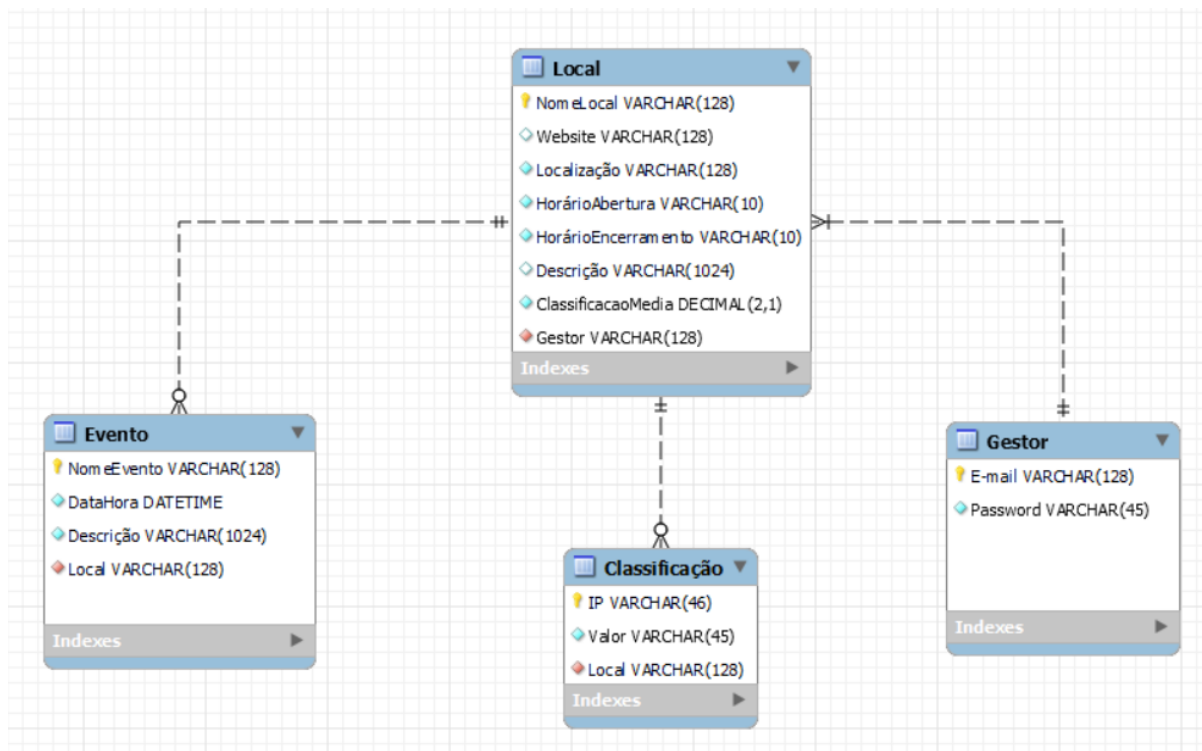


Figura 27 - Modelo lógico da Base de Dados

Descrição de relacionamentos:

Entidades

- **Fortes**: Local de Interesse (Chave primária: NomeLocal), Eventos (Chave primária: NomeEvento). Gestor (Chave primária: E-mail), Classificação (Chave Primária: IP).

Uma entidade forte é independente de outras entidades e é caracterizada pelo facto de possuir uma chave primária a ela associada de modo a identificar cada ocorrência da entidade.

- **Fracas**: A nossa base de dados não apresenta qualquer entidade fraca

Uma entidade fraca depende da existência de outras entidades e não possui chave primária.

Relacionamentos

Um Para Muitos (1:N): Gestor(1) “gere” Local(N), Local(1) “possui” Evento(N), Local(1) “possui” Classificação(N).

Uma relação de Um Para Muitos ocorre quando uma entidade de cardinalidade 1 pode conter várias ocorrências de uma outra entidade de cardinalidade(N) e cada ocorrência das mesmas apenas se pode relacionar com uma única ocorrência da entidade com cardinalidade 1, pois possui uma “cópia” da chave primária da mesma (chave estrangeira).

5. Esboço dos Interfaces do Sistema

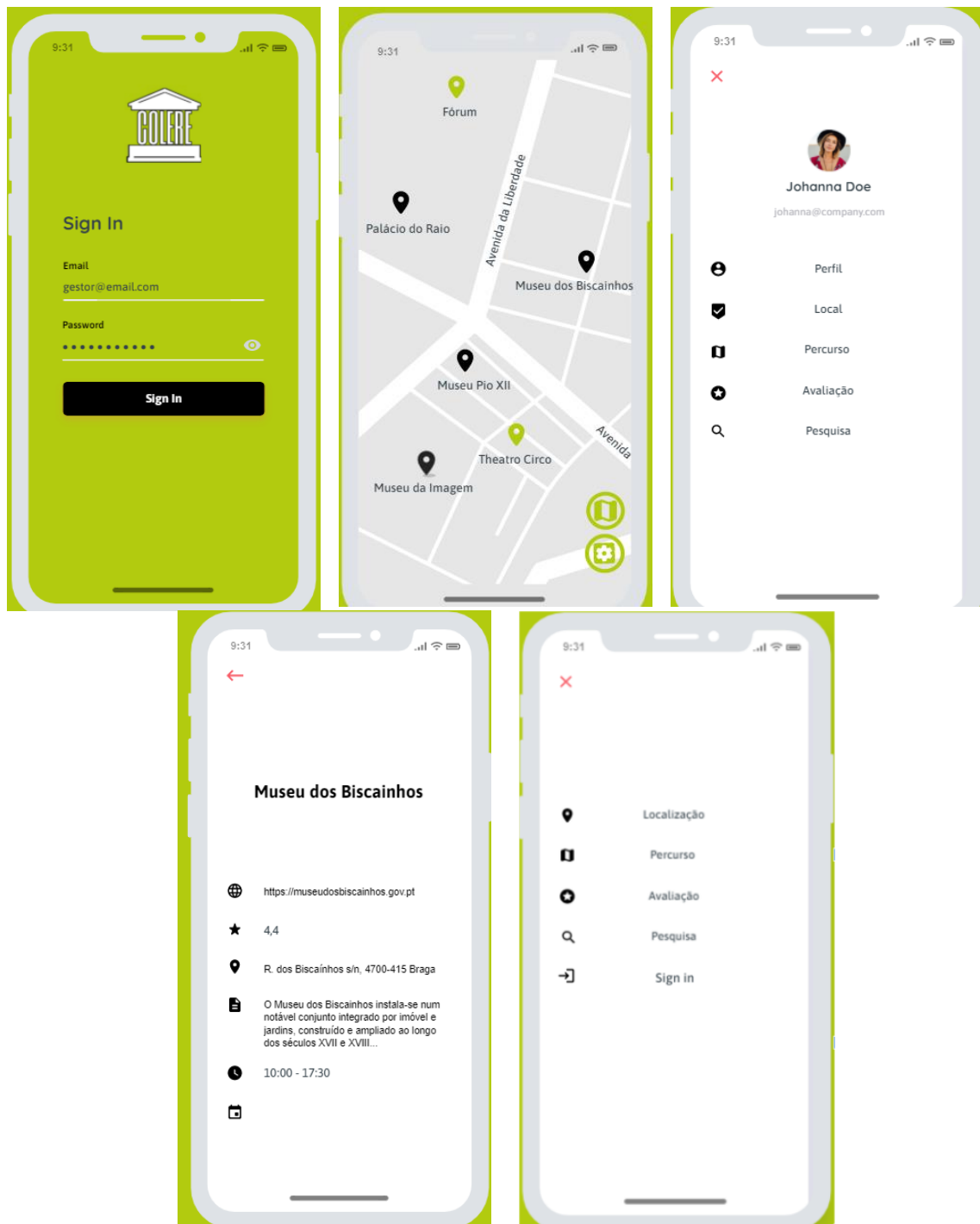


Figura 28 - Mockups

No primeiro mockup apresentado, decidimos incluir um mapa, no qual temos dois ícones. O primeiro ícone serve para criar um percurso e o segundo leva-nos para um dos menus apresentados a seguir, sendo direcionado para o primeiro caso seja um gestor autenticado e para o segundo se for um utilizador. Esta é a primeira imagem que aparecerá ao abrir a aplicação. Decidimos destacar os locais que têm eventos agendados com a cor verde, sendo os restantes apresentados a preto. Adicionalmente, cada local deve apresentar o seu nome.

As duas imagens seguintes são referentes ao gestor. Primeiro, temos a página para este se autenticar e de seguida temos uma face que só é apresentada ao gestor após este efetuar o login. Para o utilizador, temos também um menu, onde temos a opção de localização, para quando não for possível aceder à sua localização este a poder introduzir manualmente. Temos também as várias opções já apresentadas anteriormente.

Finalmente, no último mockup apresentamos um local com as respetivas informações. O último ícone é referente aos eventos, neste caso encontra-se vazia uma vez que este local não tem eventos marcados.

6. Conclusões e Trabalho Futuro

Concluída a primeira fase do desenvolvimento da aplicação Colere, pudemos constatar que o mesmo foi dividido em duas partes.

Na fundamentação, desenvolvemos de uma forma clara aquilo que desejávamos implementar e quais eram os nossos objetivos, isto é, o que pretendíamos alcançar com a aplicação.

Durante este processo, observamos, também, que a câmara de Braga e os representantes dos locais de interesse cultural se mostraram muito interessados no software que estávamos a desenvolver. Isto provou ser um ótimo incentivo para continuarmos o projeto que estamos a construir.

Na especificação, tentamos expor e analisar com mais detalhe aquilo que queríamos desenvolver, assim como os desejos dos utilizadores e interessados no sucesso da aplicação. Contudo, não seria possível obter o produto final sem que tivéssemos que comprometer certas funcionalidades por outras que achamos indispensáveis.

A análise mais detalhada permitiu-nos ter uma melhor perceção das dimensões e funcionalidades da nossa aplicação e consideramos ter desenvolvido um projeto com uma complexidade razoável, e que será possível desenvolver nos prazos impostos.

Esperamos que a implementação do projeto seja bem conseguida e reconhecemos que poderá haver necessidade de adaptar os diagramas feitos.

Tentamos desenvolver um programa que se mostrasse prático e de fácil manutenção, uma vez que, posteriormente e caso este seja bem recebido, possa ser expandido a outras áreas do turismo da cidade ou até mesmo a outras cidades.

Referências

Sommerville, I. (2016). Software Engineering. 10th ed. London: Pearson Education.