FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Bases de Dados

Maio 2019

Base de Dados de
Site de Reservas de Alojamento Local

Turma 6 - Grupo 4

João Praça - up201704748

Lucas Ribeiro - up201705227

Sílvia Rocha - up201704684

Índice

Descrição do Tema e da sua Modelação

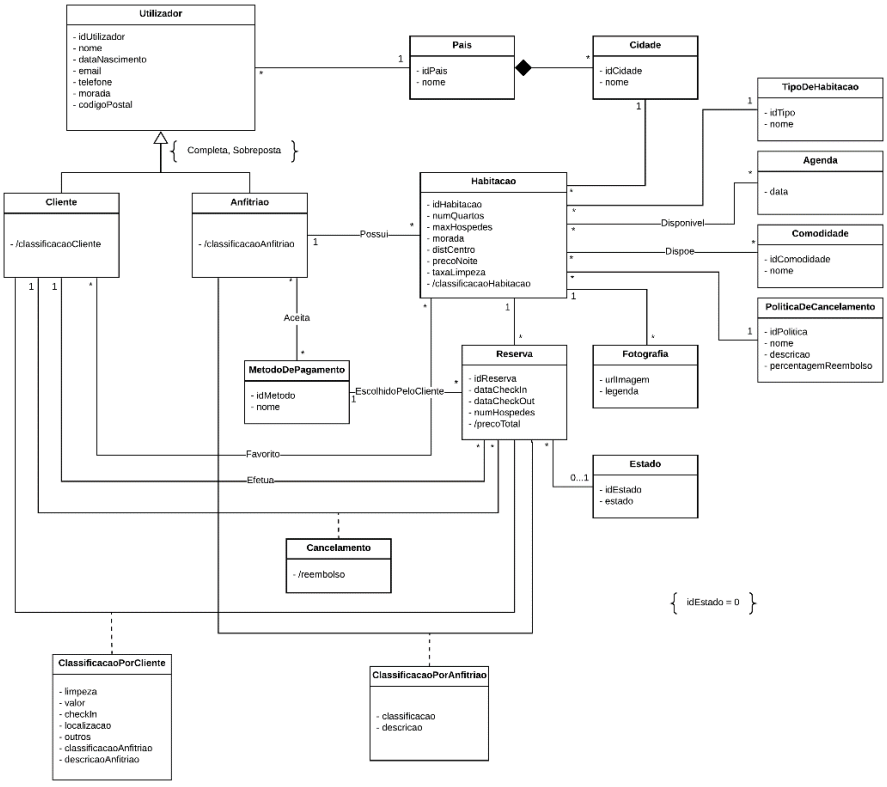
Baseado no sistema e organização da *Airbnb*, esta plataforma *online* permite aos seus clientes reservarem habitações de alojamento local fornecidas por anfitriões espalhados por todo o mundo. Aquando do registo, cada utilizador deve fornecer o seu nome, data de nascimento, e-mail, telefone, nacionalidade, morada e código postal. Após estar registado, um cliente poderá efetuar reservas e marcar habitações como favoritas. Caso o utilizador seja um anfitrião, tem ainda de escolher os métodos de pagamento que aceita. Assim que esteja registado, poderá adicionar as suas habitações. Para além disso, um utilizador pode ser um cliente e anfitrião simultaneamente.

Por sua vez, uma habitação pode ser caracterizada segundo o seu tipo (moradia, apartamento, entre outros), número de quartos, número máximo de hóspedes, morada, cidade, distância ao centro da cidade, preço por noite, disponibilidade, comodidades (ar-condicionado, Wi-Fi, entre outros), taxa de limpeza, algumas fotografias e deverá estar associada a uma política de cancelamento de reservas.

Ao realizar uma reserva, o cliente deve especificar a data de check-in, data de check-out, o método de pagamento dentro dos disponíveis e o número de hóspedes, sendo-lhe apresentado o preço total. Após concluída, a reserva fica associada a um estado da estadia (pendente, a decorrer ou concluída) e pode também ser cancelada (no caso de a estadia estar pendente), originando um reembolso de valor variável, de acordo com a política de cancelamento escolhida pelo anfitrião. No final de uma estadia, o cliente classifica a habitação em vários parâmetros (limpeza, valor, localização, entre outros) com uma pontuação de 0 a 5, e o seu anfitrião com uma pontuação geral e uma pequena descrição. O anfitrião também deverá avaliar o hóspede, mais uma vez com uma pontuação geral e uma descrição. Desta forma, o cliente, anfitrião e habitação são também caracterizados por uma classificação, que será a média aritmética das classificações que lhes foram atribuídas.

Modelo Conceptual

Com base na estrutura apresentada no ponto anterior do presente documento, obtivemos o modelo conceptual visível no diagrama UML seguinte (nos Anexos é possível ver o diagrama com mais detalhe, assim como as versões anteriores do mesmo).



Como é visível na imagem, este modelo conceptual é composto por:

- dezassete classes.
- dezanove associações de entre as quais uma é composta, duas são derivadas e três contêm classes de associação.

Esquema Relacional

Utilizador (idUtilizador, nome, dataNascimento, email, telefone, morada, codigoPostal, idPais->Pais)

Cliente (idUtilizador->Utilizador, classificacaoCliente)

Anfitriao (idUtilizador->Utilizador, classificacaoAnfitriao)

Habitacao (idHabitação, numQuartos, maxHospedes, morada, distCentro, precoNoite, taxaLimpeza, classificacaoMedia, idCidade->Cidade, idTipo->TipoDeHabitacao, idPolitica->PoliticaDeCancelamento)

Reserva (idReserva, dataCheckIn, dataCheckOut, numHospedes, precoTotal, idHabitacao->Habitacao, idEstado->Estado)

Estado (idEstado, estado)

MetodoDePagamento (idMetodo, nome)

Pais (idPais, nome)

Cidade (idCidade, nome, idPais->Pais)

TipoDeHabitacao (idTipo, nome)

Agenda (data)

Comodidade (idComodidade, nome)

PoliticaDeCancelamento (idPolitica, nome, descricao, percentagemReembolso)

Fotografia (urlImagem, legenda, idHabitacao->Habitacao)

ClassificacaoPorCliente (limpeza, valor, checkIn, localizacao, outros, classificacaoAnfitriao, descricaoAnfitriao, anfitriao->Anfitriao, reserva->Reserva)

ClassificacaoPorAnfitriao (classificacao, descricao, idCliente->Cliente, idReserva->Reserva)

Possui (anfitriao->Anfitriao, idHabitacao->Habitacao)

Favorito (cliente->Cliente, idHabitacao->Habitacao)

Efetua (cliente->Cliente, idReserva->Reserva)

Cancelamento (cliente->Cliente, idReserva->Reserva, reembolso)

Aceita (anfitriao->Anfitriao, idMetodo->MetodoDePagamento)

EscolhidoPeloCliente (idMetodo->MetodoDePagamento, idReserva->Reserva)

Disponivel (data->Agenda, idHabitacao->Habitacao)

Dispoe (idComodidade->Comodidade, idHabitacao->Habitacao)

Análise de Dependências Funcionais e Formas Normais

Um atributo, ou conjunto de atributos, é chave de uma relação quando o closure desses atributos nos permite obter todos os atributos da relação, isto é, quando todos os atributos da relação podem ser determinados funcionalmente pelo atributo inicial.

Utilizador

Utilizador (idUtilizador, nome, dataNascimento, email, telefone, morada, codigoPostal, pais->Pais)

- idUtilizador -> nome, dataNascimento, email, telefone, morada, codigoPostal, pais
- email -> idUtilizador, nome, dataNascimento, telefone, morada, codigoPostal, pais
- telefone -> idUtilizador , nome, dataNascimento, email, morada, codigoPostal, pais

Justificação: Segundo a definição mencionada, idUtilizador, email e telefone são chave de Utilizador. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

Habitacao

Habitacao (idHabitacao, numQuartos, maxHospedes, morada, distCentro, precoNoite, taxaLimpeza, classificacaoMedia, cidade->Cidade, tipo->TipoDeHabitacao, politica->PoliticaDeCancelamento)

- idHabitacao -> numQuartos, maxHospedes, morada, distCentro, precoNoite, taxaLimpeza, classificacaoMedia, cidade, tipo, politica
- morada -> idHabitacao, numQuartos, maxHospedes, distCentro, precoNoite, taxaLimpeza, classificacaoMedia, cidade, tipo, politica

Justificação: Segundo a definição mencionada, idHabitacao e morada são chaves de Habitação. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

Reserva

Reserva (idReserva, dataCheckIn, dataCheckOut, numHospedes, precoTotal, habitacao->Habitacao, estado->Estado)

- idReserva -> dataCheckIn, dataCheckOut, numHospedes, precoTotal, habitação, estado->Estado
- dataCheckIn, habitação -> idReserva, dataCheckOut, numHospedes, precoTotal, estado->Estado

Justificação: Segundo a definição mencionada, idReserva e o tuplo dataCheckIn e habitação são chaves de Reserva. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

Estado

Estado (idEstado, estado)

- idEstado-> estado
- estado -> idEstado

Justificação: Segundo a definição mencionada, idEstado e estado são chaves de Estado. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

MetodoDePagamento

MetodoDePagamento (idMetodo, nome)

- idMetodo -> nome
- nome -> idMetodo

Justificação: Segundo a definição mencionada, idMetodo e nome são chaves de MetodoDePagamento. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

Pais

Pais (idPais, nome)

- idPais -> nome
- nome -> idPais

Justificação: Segundo a definição mencionada, idPais e nome são chaves de Pais. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

Cidade

Cidade (idCidade, nome, pais->Pais)

- idCidade -> nome, pais
- nome -> idCidade, pais

Justificação: Segundo a definição mencionada, idCidade e nome são chaves de Cidade. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

TipoDeHabitacao

TipoDeHabitacao (idTipo, nome)

- idTipo -> nome
- nome -> idTipo

Justificação: Segundo a definição mencionada, idTipo e nome são chaves de TipoDeHabitacao. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

Comodidade

Comodidade (idComodidade, nome)

- idComodidade -> nome
- nome -> idComodidade

Justificação: Segundo a definição mencionada, idComodidade e nome são chaves de Comodidade. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

PoliticaDeCancelamento

PoliticaDeCancelamento (idPolitica, nome, descricao, percentagemReembolso)

- idPolitica -> nome, descricao, percentagemReembolso
- nome -> idPolitica, descricao, percentagemReembolso

Justificação: Segundo a definição mencionada, idPolitica e nome são chaves de PoliticaDeCancelamento. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

Fotografia

Fotografia (urlImagem, legenda, habitacao->Habitacao)

- urlImagem -> legenda, habitacao

Justificação: Segundo a definição mencionada, urlImagem é chave de Fotografia. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

ClassificacaoPorCliente

ClassificacaoPorCliente (limpeza, valor, checkIn, localizacao, outros, classificacaoAnfitriao, descricaoAnfitriao, anfitriao->Anfitriao, reserva->Reserva)

- reserva -> limpeza, valor, checkIn, localizacao, outros, classificacaoAnfitriao, descricaoAnfitriao, anfitriao

Justificação: Segundo a definição mencionada, reserva é chave de ClassificacaoPorCliente. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

ClassificacaoPorAnfitriao

ClassificacaoPorAnfitriao (classificacao, descricao, cliente->Cliente, reserva->Reserva)

- reserva -> descricao, classificacao, cliente

Justificação: Segundo a definição mencionada, reserva é chave de ClassificacaoPorAnfitriao. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

Possui

Possui (anfitriao->Anfitriao, habitacao->Habitacao)

- habitação -> anfitriao

Justificação: Segundo a definição mencionada, habitação é chave de Possui. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

Efetua

Efetua (cliente->Cliente, reserva->Reserva)

- reserva -> cliente

Justificação: Segundo a definição mencionada, reserva é chave de Efetua. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

Cancelamento

Cancelamento (cliente->Cliente, reserva->Reserva, reembolso)

- reserva -> reembolso, cliente

Justificação: Segundo a definição mencionada, reserva é chave de Cancelamento. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

EscolhidoPeloCliente

EscolhidoPeloCliente (metodo->MetodoDePagamento, reserva->Reserva)

- reserva -> metodo

Justificação: Segundo a definição mencionada, reserva é chave de EscolhidoPeloCliente. Sendo que todos os atributos do lado esquerdo das dependências funcionais são chaves, a relação está na BCNF e consequentemente na 3rd NF.

Outras Relações

As restantes relações não apresentam dependências funcionais não triviais.

Restrições da Base de Dados

Agenda

Não pode haver duas datas repetidas (restrição PRIMARY KEY).

Pais

Não pode haver dois países com o mesmo id (restrição PRIMARY KEY).

Não pode haver países com nome nulo (restrição NOT NULL).

Não pode haver dois países com o mesmo nome (restrição UNIQUE).

Cidade

Não pode haver duas cidades com o mesmo id (restrição PRIMARY KEY).

Não pode haver cidades com nome nulo (restrição NOT NULL).

Não pode haver cidades associadas a um país inexistente (restrição de integridade referencial).

Utilizador

Não pode haver dois utilizadores com o mesmo id (restrição PRIMARY KEY).

Não pode haver utilizadores com nome nulo (restrição NOT NULL).

Não pode haver utilizadores com data de nascimento nula (restrição NOT NULL).

Não pode haver utilizadores com e-mail nulo (restrição NOT NULL).

Não pode haver dois utilizadores com o mesmo e-mail (restrição UNIQUE).

Não pode haver utilizadores com telefone nulo (restrição NOT NULL).

Não pode haver dois utilizadores com o mesmo telefone (restrição UNIQUE).

Não pode haver utilizadores com morada nula (restrição NOT NULL).

Não pode haver utilizadores com código postal nulo (restrição NOT NULL).

Não pode haver utilizadores associados a um país inexistente (restrição de integridade referencial).

Cliente

Não pode haver clientes associados a um utilizador inexistente (restrição de integridade referencial).

Não pode haver clientes com classificação menor que 1 ou maior que 5 (restrição CHECK).

Anfitriao

Não pode haver anfitriões associados a um utilizador inexistente (restrição de integridade referencial).

Não pode haver anfitriões com classificação menor que 1 ou maior que 5 (restrição CHECK).

MetodoDePagamento

Não pode haver dois métodos de pagamento com o mesmo id (restrição PRIMARY KEY).

Não pode haver dois métodos de pagamento com o mesmo nome (restrição UNIQUE).

Não pode haver métodos de pagamento com nome nulo (restrição NOT NULL).

Aceita

Não pode haver anfitriões associados a um anfitrião inexistente (restrição de integridade referencial).

Não pode haver métodos associados a um método inexistente (restrição de integridade referencial).

Não pode haver duas combinações de anfitriões e métodos iguais (restrição PRIMARY KEY).

Reserva

Não pode haver duas reservas com o mesmo id (restrição PRIMARY KEY).

Não pode haver reservas com data de check-in nula (restrição NOT NULL).

Não pode haver reservas com data de check-out nula (restrição NOT NULL).

Não pode haver reservas com número de hóspedes menor que 1 (restrição CHECK).

Não pode haver reservas com preço total menor ou igual a 0 (restrição CHECK).

Não pode haver reservas associadas a uma habitação inexistente (restrição de integridade referencial).

Não pode haver duas combinações de data de check-in e habitação iguais (restrição UNIQUE).

Não pode haver reservas associadas a um estado inexistente (restrição de integridade referencial).

Estado

Não pode haver dois estados com o mesmo id (restrição PRIMARY KEY).

Não pode haver estados com estado nulo (restrição NOT NULL).

Não pode haver dois estados com o mesmo atributo estado (restrição UNIQUE).

Cancelamento

Não pode haver cancelamentos com reembolso nulo (restrição NOT NULL).

Não pode haver cancelamentos associados a um cliente inexistente (restrição de integridade referencial).

Não pode haver cancelamentos associados a uma reserva inexistente (restrição de integridade referencial).

Não pode haver dois cancelamentos para a mesma reserva (restrição PRIMARY KEY).

ClassificacaoPorAnfitriao

Não pode haver classificações menor que 1 e maior que 5 (restrição CHECK).

Não pode haver classificações associadas a um anfitrião inexistente (restrição de integridade referencial).

Não pode haver classificações associadas a uma reserva inexistente (restrição de integridade referencial).

Não pode haver duas classificações para a mesma reserva (restrição PRIMARY KEY).

ClassificacaoPorCliente

Não pode haver limpeza menor que 1 e maior que 5 (restrição CHECK).

Não pode haver valor menor que 1 e maior que 5 (restrição CHECK).

Não pode haver checkIn menor que 1 e maior que 5 (restrição CHECK).

Não pode haver localização menor que 1 e maior que 5 (restrição CHECK).

Não pode haver classificacaoAnfitriao menor que 1 e maior que 5 (restrição CHECK).

Não pode haver classificações associadas a um cliente inexistente (restrição de integridade referencial).

Não pode haver classificações associadas a uma reserva inexistente (restrição de integridade referencial).

Não pode haver duas classificações para a mesma reserva (restrição PRIMARY KEY).

Comodidade

Não pode haver duas comodidades com o mesmo id (restrição PRIMARY KEY).

Não pode haver comodidades com nome nulo (restrição NOT NULL).

Não pode haver duas comodidades com nome igual (restrição UNIQUE).

Efetua

Não pode haver clientes associados a um cliente inexistente (restrição de integridade referencial).

Não pode haver reservas associados a uma reserva inexistente (restrição de integridade referencial).

Não pode haver duas combinações de clientes e reservas iguais (restrição PRIMARY KEY).

EscolhidoPeloCliente

Não pode haver métodos associados a um método inexistente (restrição de integridade referencial).

Não pode haver reservas associados a uma reserva inexistente (restrição de integridade referencial).

Não pode haver duas combinações de métodos e reservas iguais (restrição PRIMARY KEY).

TipoDeHabitacao

Não pode haver dois tipos de habitação com o mesmo id (restrição PRIMARY KEY).

Não pode haver um tipo de habitação com nome nulo (restrição NOT NULL).

Não pode haver dois tipos de habitação com nome igual (restrição UNIQUE).

PoliticaDeCancelamento

Não pode haver duas políticas com o mesmo id (restrição PRIMARY KEY).

Não pode haver uma política com nome nulo (restrição NOT NULL).

Não pode haver duas políticas com nome igual (restrição UNIQUE).

Não pode haver uma política com descrição nula (restrição NOT NULL).

Não pode haver duas políticas com descrição igual (restrição UNIQUE).

A percentagem de reembolso não pode ser inferior a 0 ou superior a 1 (restrição CHECK).

Habitacao

Não pode haver duas habitações com o mesmo id (restrição PRIMARY KEY).

O número de quartos não pode ser inferior a 0 (restrição CHECK).

O número máximo de hóspedes não pode ser inferior a 0 (restrição CHECK).

Não pode haver uma habitação com morada nula (restrição NOT NULL).

Não pode haver duas habitações com morada igual (restrição UNIQUE).

A distância ao centro não pode ser inferior a 0 (restrição CHECK).

O preço por noite não pode ser inferior ou igual a 0 (restrição CHECK).

A taxa de limpeza não pode ser inferior a 0 (restrição CHECK).

O valor da classificação média não pode ser inferior a 1 nem superior a 5 (restrição CHECK).

Não pode haver cidades associadas a uma cidade inexistente (restrição de integridade referencial).

Não pode haver tipos associados a um tipo de habitação inexistente (restrição de integridade referencial).

Não pode haver políticas associadas a uma política de cancelamento inexistente (restrição de integridade referencial).

Disponivel

Não pode haver datas associadas a uma data inexistente (restrição de integridade referencial).

Não pode haver habitações associadas a uma habitação inexistente (restrição de integridade referencial).

Não pode haver duas combinações de datas e habitações iguais (restrição PRIMARY KEY).

Dispoe

Não pode haver comodidades associadas a uma comodidade inexistente (restrição de integridade referencial).

Não pode haver habitações associadas a uma habitação inexistente (restrição de integridade referencial).

Não pode haver duas combinações de comodidades e habitações iguais (restrição PRIMARY KEY).

Favorito

Não pode haver clientes associados a um cliente inexistente (restrição de integridade referencial).

Não pode haver habitações associadas a uma habitação inexistente (restrição de integridade referencial).

Não pode haver duas combinações de clientes e habitações iguais (restrição PRIMARY KEY).

Fotografia

Não pode haver duas fotografias com o mesmo urlImagem(restrição PRIMARY KEY).

Não pode haver habitações associadas a uma habitação inexistente (restrição de integridade referencial).

Possui

Não pode haver anfitriões associados a um anfitrião inexistente (restrição de integridade referencial).

Não pode haver habitações associadas a uma habitação inexistente (restrição de integridade referencial).

Não pode haver duas combinações de anfitriões e habitações iguais (restrição PRIMARY KEY).

Interrogação da Base de Dados

1. **Quais os dias disponíveis de cada habitação.** Esta interrogação lista todas as datas em que, cada uma das habitações presente na base se dados, se encontra disponível para ser reservada. De notar que para que esta interrogação funcione corretamente é necessário que, anteriormente, tenha sido adicionado o gatilho 3, que será descrito mais à frente, para que não sejam listadas datas em que já existem reservas.

2. **Quais as habitações do anfitrião cujo id é 1.** Esta interrogação lista todas as habitações listadas como pertencentes ao anfitrião 1.

3. **Quais as habitações da cidade com id 7 (Portimão) que o cliente com id 1 já visitou.** Esta interrogação irá listar todas as reservas já concluídas pelo cliente 1 na cidade de Portimão.

4. **Quais as habitações, ordenadas pelo preço mais baixo, pertencentes a Portimão, disponíveis entre 25 e 30 de julho de 2019 e com capacidade mínima de 2 pessoas.** Esta interrogação irá listar todas as habitações que cumprem todas as restrições mencionadas, ordenando-as pelo valor total mais baixo (preço por noite multiplicado pelo número de noites, somado com a taxa de limpeza a aplicar).

5. **Quais as classificações da habitação com id 1.** Esta interrogação irá listar todas as classificações desta habitação mostrando os vários parâmetros da classificação que se aplicam à habitação.

6. **Qual a moda das classificações da habitação com id 6.** Esta interrogação irá calcular qual a moda, isto é, a classificação mais frequente, de cada um dos parâmetros de avaliação da habitação para a habitação de id 6.

7. **Qual a cidade mais reservada em cada mês.** Esta interrogação lista, para cada mês em que existem reservas, qual a cidade mais reservada nessas datas. Essa listagem mostra o mês e cidade e é ordenada segundo o mês.

8. **Quais as reservas pendentes do cliente 1.** Esta interrogação lista todas as reservas efetuadas pelo cliente 1 e que se encontram pendentes, isto é, que não foram canceladas mas cuja data de Check-In ainda não foi atingida.

9. **Quais os pares de habitações semelhantes.** Esta interrogação lista pares de habitações semelhantes, isto é, pares de habitações com o mesmo número de quartos, tipo de habitação e localizadas na mesma cidade.

10. **Quais os clientes que cancelaram todas as reservas.** Esta interrogação lista todos os clientes que cancelaram todas as suas reservas, isto é, que efetuaram pelo menos uma reserva mas que nunca concluíram nenhuma nem têm nenhuma pendente.

Descrição dos gatilhos implementados

- **Gatilho 1 -** Tem como objetivo impedir que um cliente classifique uma habitação, e respetivo anfitrião, antes que haja uma reserva concluída associada a esse mesmo cliente e habitação. O gatilho atinge este objetivo ao ignorar qualquer operação INSERT na classificação de anfitrião que seja efetuada, antes que o idEstado esteja igual a 0 (id correspondente ao estado "Concluido").

  Para o correto funcionamento da base de dados, seria ainda necessário criar um outro gatilho com objetivo semelhante para operações UPDATE do estado e da classificação.
- **Gatilho 2 -** Tem como objetivo atualizar a classificação dos anfitriões com a média das classificações ao anfitrião pela totalidade dos clientes, tal como atualizar a classificação das habitações com a média das médias das classificações atribuídas pelos clientes a cada critério da classificação de uma habitação. O gatilho atinge este objetivo ao fazer as atualizações referidas depois de cada tentativa de inserção de novos dados na relação de classificação por cliente.

  Para o correto funcionamento da base de dados, seria ainda necessário criar outros gatilhos com objetivo semelhante para operações UPDATE e DELETE de classificações.
- **Gatilho 3 -** Tem como objetivo atualizar a disponibilidade de uma casa em datas para as quais são efetuadas reservas. Para atingir este objetivo, o gatilho apaga todas as

relações de disponibilidade, entre as datas de check-In e check-out, com id igual ao da habitação à qual se tentou efetuar uma operação INSERT na reserva.

Para o correto funcionamento da base de dados, seria ainda necessário criar outros gatilhos com objetivo semelhante para operações UPDATE e DELETE de reservas, isto é, caso as datas de check-In e check-Out da reserva sejam alteradas ou a reserva seja apagada (este gatilho aplicar-se-ia também em casos de cancelamento da reserva).

Para além do que já foi mencionado acima, para o correto funcionamento da base de dados, seria ainda necessária a implementação de, pelo menos, mais quatro gatilhos com os propósitos a seguir descritos:

- **Impedir que um anfitrião classifique um cliente antes que a reserva esteja concluída.** Este gatilho teria de ser implementado para operações INSERT da classificacaoPorAnfitriao. O seu funcionamento seria muito semelhante ao já descrito para a classificacaoPorCliente (gatilho 1).
- **Atualizar a classificação de um cliente sempre que um anfitrião o classifica**. Este gatilho teria de ser implementado para operações INSERT, UPDATE e DELETE de classificacaoPorAnfitriao. O seu funcionamento seria muito semelhante ao já descrito para a classificacaoPorCliente (gatilho 2).
- **Atualizar o valor do reembolso de um Cancelamento.** Este gatilho teria de ser implementado para operações INSERT de Cancelamento, operações DELETE da Reserva e operações UPDATE do Estado. O seu funcionamento consistiria em sempre que alguma das operações mencionadas fosse executada, o valor do Cancelamento era atualizado usando o atributo precoTotal da reserva e a percentagemReembolso da política de cancelamento associada à habitação a que a reserva a cancelar corresponde.
- **Atualizar o preço total de uma reserva.** Este gatilho teria de ser implementado para operações INSERT e UPDATE de uma Reserva e operações UPDATE de uma habitação. O seu funcionamento consistiria em sempre que uma nova reserva fosse criada ou os atributos dataCheckIn ou dataCheckOut (da reserva) ou precoNoite ou taxaLimpeza (da Habitação) fossem alterados, o valor do precoTotal também o fosse.

Anexos

Diagrama UML da primeira entrega

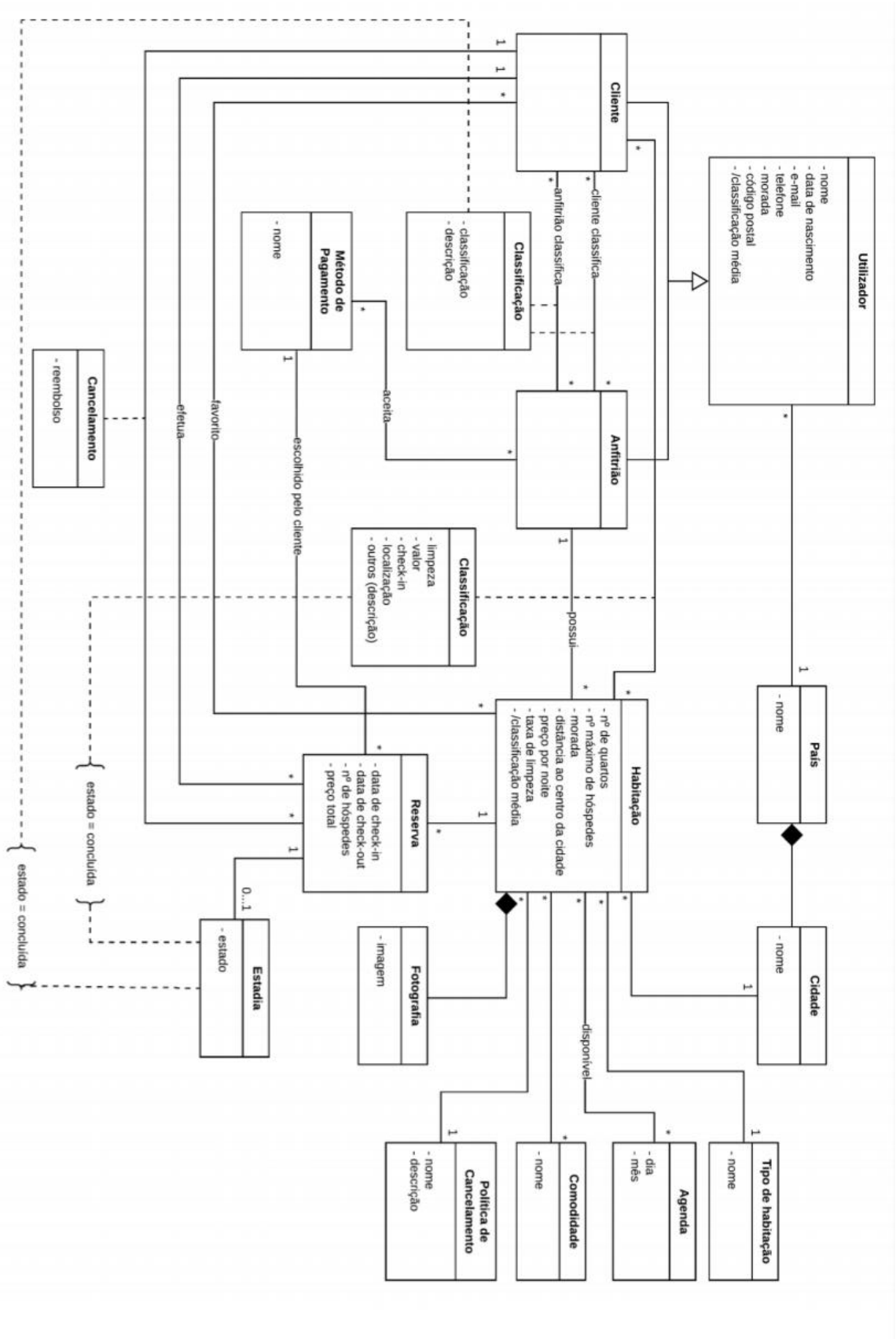


Diagrama UML da segunda entrega

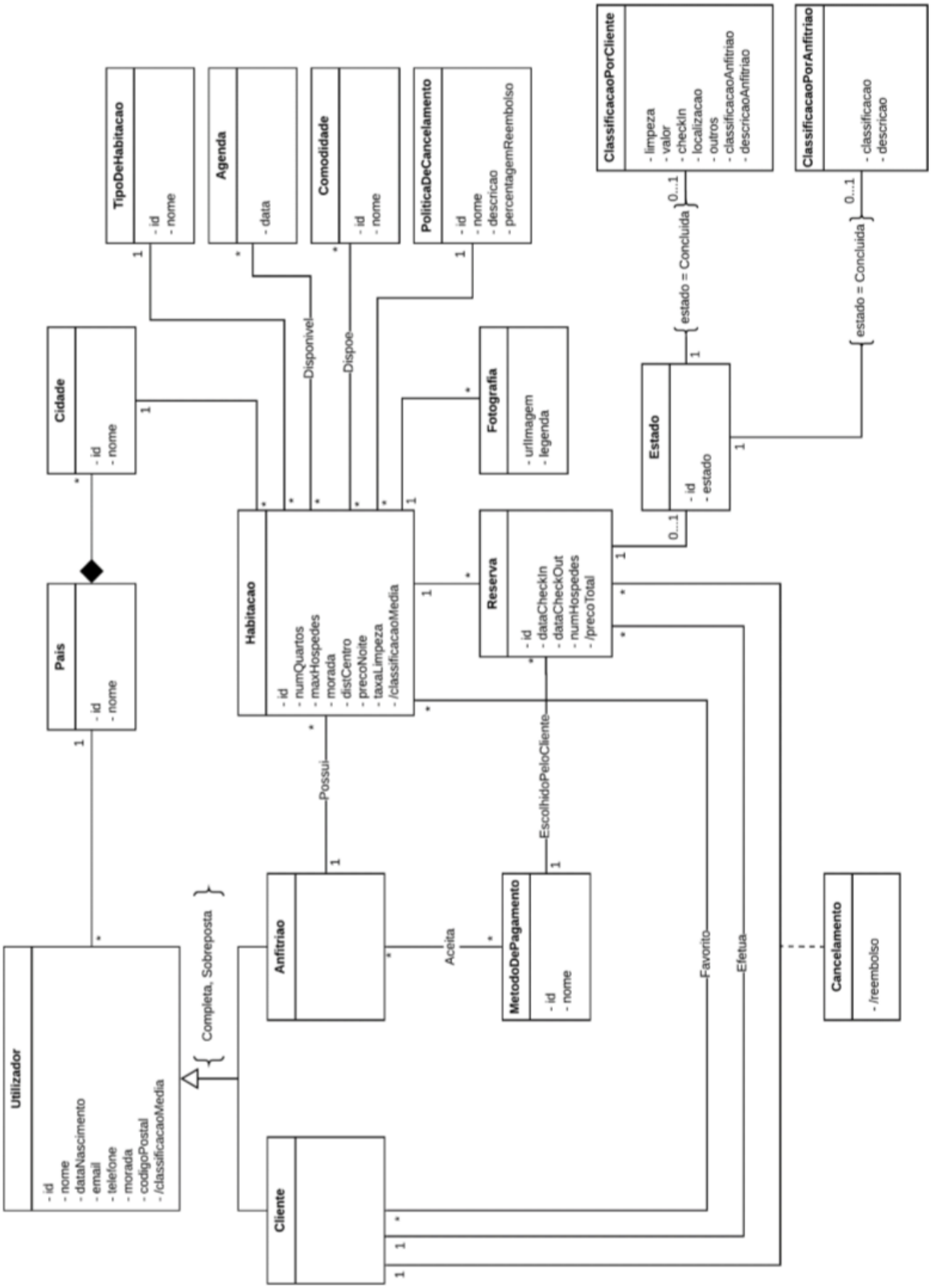


Diagrama UML final