

Decisões de Projeto do MER

Artur De Vlieger Lima [13671574], Gabriel da Costa Merlin [12544420],
Gustavo Gabriel Ribeiro [13672683], Lucas de Souza Brandão [13695021]

1. Conjuntos de Entidades

Aqui, abordaremos quais são as entidades, e seus atributos, que usamos para modelar o problema proposto no Diagrama de Entidade Relacionamento.

a. Conjuntos de Entidades Regulares

- Propriedade: Representa os imóveis ou quartos disponibilizados no sistema para locação.
 - Atributos: nome, endereco, tipo, qtd_quartos, qtd_banheiros, max_hospedes, min_noites, max_noites, preco_noite, taxa_limpeza, data_inicio, data_fim, horario_checkIn, horario_checkOut, regras, comodidades
- Localização: Traz informações a respeito de onde mora um usuário (anfitrião ou locatário) e onde se localiza uma propriedade.
 - Atributos: id_local, pais, estado, cidade, bairro, pts_interesse.

b. Conjuntos de Entidades Fracas

- Conta: Informações importantes sobre a conta de anfitriões.
 - Atributos: num_conta, num_roteamento, tipo_conta.
- Quarto: Possível complemento de propriedade, trazendo informações adicionais que podem vir a ser importantes para o locatário em busca de uma propriedade.
 - Atributos: id_quarto, qtd_camras, tipo_camras.

c. Abstração de Generalização

- Usuário: Conjunto de Entidade Genérica. Representa todos os usuários do sistema
 - Atributos: nome, sobrenome, num_tel, data_nasc, endereco, sexo, email, senha
- Anfitrião: Especialização de Usuário. Representa os usuários que possuem uma propriedade.
 - Atributos específicos: NÃO HÁ
- Locatário: Especialização de Usuário. Representa os usuários que buscam locar uma propriedade (ser locatário não exclui a possibilidade de ser anfitrião e vice-versa).
 - Atributos específicos: NÃO HÁ

d. Abstração de Agregação

- **Locação:** Abstração de Agregação. Armazena informações sobre a locação de uma propriedade realizada por um locatário.
 - Atributos: data_checkIn +, data_checkOut, data_reserva, qtd_hospedes, imposto, preco_estadia, cod_promocional, desconto, preco_total, condicao
- **Avaliação:** Armazena informações relevantes acerca da avaliação feitas por locatários em relação a propriedades.
 - Atributos: tempo +, fotos, rate_limpeza, rate_comunicacao, rate_localizacao, mensagem

2. Conjunto de Relacionamentos

Agora, mostraremos quais os relacionamentos pensados entre entidades propostos por nós.

- **Ter quarto:** Relaciona a propriedade aos quartos que ela possui, uma propriedade podendo possuir N quartos.
- **Localizar propriedade:** Relaciona uma propriedade à sua localização em uma relação 1:N.
- **Localizar Usuário:** Relaciona um usuário à sua localização em um relacionamento 1:N.
- **Possuir:** Relaciona o anfitrião às propriedades que ele possui, um anfitrião podendo possuir N propriedades.
- **Locar:** Relaciona um locatário a uma propriedade que ele reservou em um relacionamento M:N. Esse relacionamento gera a Abstração de Agregação *Locação*.
- **Avaliar:** Relaciona M locatários a N propriedades, possibilitando que eles avaliem propriedades. Esse relacionamento gera a Abstração de Agregação *Avaliação*.
- **Refere-se a:** Relaciona Avaliação e Locação com cardinalidade 1:1, dado que, quando um locatário avalia uma propriedade, essa avaliação, no fim das contas, refere-se à locação que ele fez dessa propriedade.
- **Ter conta:** Relaciona o anfitrião às contas que ele possui, um anfitrião podendo possuir N contas.

3. Decisões de Projetos

Durante esta seção, discutiremos as escolhas feitas pelo grupo durante a modelagem a fim de melhorar a compreensão da nossa modelagem.

- **Divisão do usuário:** Notamos que locatários e anfitriões, apesar de não terem nenhum atributo diferente um do outro, possuíam diferenças o suficiente, em termos de relacionamentos diferentes que cada um realizava, para justificar uma especialização do usuário em locatário e anfitrião.

- ‘Regras’, ‘Comodidades’ e ‘Quarto’: Tínhamos pensado em suprimir essas entidades e colocar seus atributos todos em ‘Propriedade’. No entanto, ‘Quarto’ era muito mais complexo que ‘Regras’ e ‘Comodidades’, dado que cada quarto tem seus próprios atributos, enquanto que a regras e comodidades tem como atributos apenas seus nomes. Dessa forma, ‘Quarto’ foi representado como uma entidade fraca de Propriedade e ‘Regras’ e ‘Comodidades’ como atributos multivalorados de Propriedade.
- ‘Localização’ como entidade: Em nosso primeiro rascunho, modelamos a localização como um atributo composto de ‘Locatário’, ‘Anfitrião’ e ‘Propriedade’, mas percebemos, depois, que isso causaria muita repetição de valores, além de que, dessa forma, não conseguiremos representar os pontos de interesse, que foram modelados como atributo multivalorado de Localização.
- ‘Conta’ como entidade fraca: Achamos importante torná-la uma entidade fraca pois consideramos desnecessário manter informações da conta de um anfitrião do sistema caso o anfitrião seja removido.
- Abstração de Agregação ‘Avaliação’: dado que a uma avaliação acontece fruto do relacionamento entre um locatário e uma propriedade e que é necessário identificar cada uma dessas avaliações, foi natural agregar o relacionamento ‘Avaliar’ na agregação ‘Avaliação’. Entretanto, semanticamente, essa estrutura ficou incompleta, pois, na realidade, um locatário avalia sua estadia em uma propriedade, e não somente a propriedade em si. Assim, surgiu a necessidade de criar um relacionamento 1:1 entre ‘Avaliação’ e ‘Locação’.
- Abstração de Agregação ‘Locação’: da mesma maneira que aconteceu no item anterior, a locação surge do relacionamento entre um locatário e uma propriedade, sendo necessário identificar cada um desses relacionamentos. Assim, o relacionamento ‘Locar’ foi agregado em ‘Locação’.
- ‘Mensagem’: foi entendido, pelo grupo, que não existe chat no site. Assim, mensagem é tão somente um atributo da avaliação, não sendo necessário representá-la como Entidade ou Relacionamento separadamente.
- Atributos ‘data_inicio’ e ‘data_fim’ de ‘Propriedade’: para representar as datas em que uma propriedade está disponível para locação, foi criado um intervalo de datas, a partir das datas de início da disponibilização até seu fim.
- ‘data_checkIn’ como chave adicional de ‘Locação’: entendemos que, tanto a data de check-in quanto de check-out possuem a mesma carga semântica no que se refere à identificação do CEA Locação. Assim, fica livre, sob nossa interpretação, a escolha de uma das duas para ser chave adicional. No caso, optamos por ‘data_checkIn’.

4. Regras não contempladas

Por fim, contemplaremos tudo aquilo com que o sistema deve se preocupar, mas que não é possível de se modelar usando um Diagrama de Entidade Relacionamento.

- Uma reserva pode ser confirmada ou cancelada até uma data anterior à data do check-in.
- Cada propriedade pode ter várias reservas, mas o período de uma reserva (entre as datas de check-in e check-out) não pode se sobrepor aos períodos das demais.