

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. Conjunto de Entidades

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

- **Anfitrião:** Representa os usuários que possuem uma propriedade.
 - Atributos: nome, sobrenome, num_tel, data_nasc, endereco, genero, email, senha.
- **Locatário:** Representa os usuários que buscam locar uma propriedade (ser locatário não exclui a possibilidade de ser anfitrião).
 - Atributos: nome, sobrenome, num_tel, data_nasc, endereco, genero, email, senha.
- **Propriedade:** Representa os imóveis ou quartos disponibilizados no sistema para locação.
 - Atributos: nome, endereco, tipo_propriedade, qtd_quartos, qtd_banheiros, max_hospedes, min_noites, max_noites, preco_noite, taxa_limpeza, datas_disponivel, horario_checkIn, horario_checkOut.
- **Locação:** Armazena informações sobre a locação de uma propriedade realizada por um locatário.
 - Atributos: id_locacao, imposto, qtd_hospedes, preco_estadia, data_checkIn, data_checkOut, data_reserva, cod_promocional, desconto, preco_total, condicao.
- **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.
- **Mensagem:** Armazena informações relevantes sobre a troca de mensagens entre usuários (anfitrião ou locatário).
 - Atributos: nome_envia, sobrenome_envia, num_tel_envia, horario_envio, nome_recebe, sobrenome_recebe, num_tel_recebe, texto_mensagem.
- **Avaliação:** Armazena informações relevantes acerca da avaliação feitas por locatários em relação a propriedades.
 - Atributos: id_avaliacao, fotos, rate_limpeza, rate_comunicacao, rate_localizacao, rate_valor.
- **Conta (Entidade Fraca):** Informações importantes sobre a conta de anfitriões.
 - Atributos: num_conta, num_roteamento, tipo_conta.

- **Pontos de Interesse (Entidade Fraca):** Possível complemento para a localização, trazendo mais informações sobre a localização.
 - Atributos: id_ptnInteresse, nome_ponto.
- **Regras (Entidade Fraca):** Possível complemento de propriedade, trazendo informações adicionais que podem vir a ser importantes para o locatário em busca de uma propriedade.
 - Atributos: id_regra, nome_regra.
- **Comodidade (Entidade Fraca):** Possível complemento de propriedade, trazendo informações adicionais que podem vir a ser importantes para o locatário em busca de uma propriedade.
 - Atributos: id_comodidade, nome_comodidade.
- **Quarto (Entidade Fraca):** 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_camas, tipo_camas.

2. Conjunto de Relacionamentos

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

- **Ter regras:** Relaciona a propriedade às regras que ela possui, uma propriedade podendo possuir N regras.
- **Ter comodidade:** Relaciona a propriedade às comodidades que ela possui, uma propriedade podendo possuir N comodidades .
- **Ter quarto:** Relaciona a propriedade aos quartos que ela possui, uma propriedade podendo possuir N quartos.
- **Possuir:** Relaciona o anfitrião às propriedades que ele possui, um anfitrião podendo possuir N propriedades.
- **Localizar propriedade:** Relaciona uma propriedade à sua localização em uma relação 1:N.
- **Localizar locatário:** Relaciona um locatário à sua localização em uma relação 1:N.
- **Localizar anfitrião:** Relaciona um anfitrião à sua localização em uma relação 1:N.
- **Ter ponto de interesse:** Relaciona uma localização aos seus possíveis pontos de interesse (0 a N).
- **Ter conta:** Relaciona o anfitrião às contas que ele possui, um anfitrião podendo possuir N contas.

- **Referir:** Serve para relacionar uma locação com sua respectiva propriedade que está sendo alugada. Uma propriedade pode possuir N locações.
- **Reservar:** Serve para relacionar uma locação com seu respectivo locatário que a fez. Um locatário pode possuir N locações.
- **Avaliar:** Relaciona uma avaliação, quem fez esta avaliação (locatário) e para quem (propriedade). Na cardinalidade nesse caso, 1 locatário e 1 propriedade podem ter N avaliações entre si, enquanto que 1 locatário e 1 avaliação só podem se referir a 1 propriedade e 1 propriedade e 1 avaliação só podem ter vindo de 1 locatário.
- **Enviar:** Conecta uma mensagem, quem a escreveu (locatário ou anfitrião - chamaremos de *sender*) e para quem (locatário ou anfitrião - chamaremos de *receiver*). A cardinalidade nesse caso, 1 *sender* e 1 *receiver* podem ter N mensagens entre si, enquanto que 1 *sender* e 1 mensagem só podem se referir a um *receiver*, e 1 *receiver* e 1 mensagem só podem ter vindo de 1 *sender*.
- **Conter:** Relaciona uma avaliação com a mensagem que esta avaliação pode vir a conter. A relação é 1:1.

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 possuíam diferenças o suficiente, em termos de atributos e de relações diferentes que cada um realizava, para justificar uma separação do usuário locatário e usuário anfitrião.
- **Relacionamento ‘Enviar’ como triplo:** Pensamos bastante no melhor jeito de modelar esse relacionamento, tentando encaixar vários relacionamentos binários. No entanto, estávamos sempre perdendo informações relevantes ao problema, o que nos fez escolher o relacionamento triplo.
- **‘Regras’, ‘Comodidade’ e ‘Quarto’ como entidades:** Tínhamos pensado em suprimir essas entidades e colocar seus atributos todos em ‘Propriedade’. No entanto, percebemos que cada uma delas trazia consigo em seus atributos muito valor semântico que poderia acarretar num sobrecarregamento de ‘Propriedade’, por isso, decidimos criar essas 3 entidades.
- **‘Localização’ como entidade:** Em nosso primeiro rascunho, modelamos a localização como um atributo composto de ‘Locatário’,

‘Anfitrião’ e ‘Propriedade’, mas percebemos, depois, que isso supriria esses valores (por ser atributo composto) do Modelo Relacional, algo que afetaria muito nosso modelo, já que a localização é algo muito importante para o sistema proposto. Com isso, decidimos colocar como entidade a ‘Localização’.

- **Reserva e ‘Locação’ são sinônimos:** Uma das primeiras dificuldades do grupo foi quanto à diferenciação dos conceitos de Reserva e Locação, já que o texto de apoio ao trabalho fala sobre ideias muito próximas utilizando ambas. Nossa primeira ideia foi a de termos duas entidades, uma para cada palavra. No entanto, por as duas serem muito similares acabamos por deixar apenas modelado a ‘Locação’ e suprimir a Reserva no relacionamento ‘Reservar’. Na antiga modelagem, tínhamos colocado a entidade Reserva apenas para tratar os casos nos quais elas são canceladas. Essas situações são agora regularizadas pelo atributo ‘condição’ de ‘Locação’.
- **Relacionamento ‘Avaliar’ como triplo:** Assim como o relacionamento ‘Enviar’, percebemos que usar diversos relacionamentos binários em vez de apenas um triplo (‘Avaliar’) trazia muita perda de informação.
- **‘Quarto’ como fraca:** Achamos importante torná-la uma entidade fraca pois consideramos importante remover desnecessário possuir quartos de uma propriedade não mais presente no sistema.
- **‘Regras’ como fraca:** Achamos importante torná-la uma entidade fraca pois consideramos importante remover desnecessário possuir regras de uma propriedade não mais presente no sistema.
- **‘Comodidades’ como fraca:** Achamos importante torná-la uma entidade fraca pois consideramos importante remover desnecessário possuir comodidades de uma propriedade não mais presente no sistema.
- **‘Pontos de interesse’ como fraca:** Achamos importante torná-lo uma entidade fraca pois consideramos desnecessário possuir informações de pontos de interesse que tenham sido removidos do sistema.
- **‘Conta’ como fraca:** Achamos importante torná-la uma entidade fraca pois consideramos importante remover informações da conta de um anfitrião do sistema caso o anfitrião seja removido.

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.