

Um Projeto de dba para a pousada céu azul

Hospedagem em Nuvem

Por:

Pedro Magalhães

Brasília - 2023

Cliente: Pousada Céu Azul

# Agradecimentos

Gostaria de expressar meus sinceros agradecimentos àqueles que fizeram parte, mesmo que de maneira indireta, deste projeto.

Primeiramente, meus agradecimentos ao SENAI e ao professor Luciano Lopes, que me ofertaram oportunidade de aprendizado e material de estudo. Sem o gatilho inicial de ambos, não estaria chegando tão longe.

Agradeço também às plataformas de ensino virtual: Power Tuning e Hashtag Treinamentos. Aquela pelas aulas de “Dia a Dia de um DBA”, que me ensinaram a operar o sistema SQL SERVER. E esta pelas aulas de construção de Dashboards e codificação em Python, que usarei mais tarde neste projeto e em projetos futuros na vida.

Obrigado a todos os que me ajudaram nesse caminho gratificante do aprendizado. A ajuda de todos foram significativas e essenciais.

Sumário

[Agradecimentos 2](#_Toc143531335)

[Introdução 5](#_Toc143531336)

[Propósito do projeto 5](#_Toc143531337)

[Escopo do projeto 5](#_Toc143531338)

[Descrição do Cliente 6](#_Toc143531339)

[Resumo da empresa 6](#_Toc143531340)

[Hóspedes e Acomodações 6](#_Toc143531341)

[Levantamento de Requisitos 7](#_Toc143531342)

[Regras de negócio 7](#_Toc143531343)

[Lista de entidades 7](#_Toc143531344)

[Associativas 7](#_Toc143531345)

[Lista de atributos 7](#_Toc143531346)

[Associativas 7](#_Toc143531347)

[Modelo Conceitual 8](#_Toc143531348)

[Modelo Lógico 9](#_Toc143531349)

[Banco de Dados 10](#_Toc143531350)

[Schemas 10](#_Toc143531351)

[Tabelas 11](#_Toc143531352)

[Telefone\_Pessoa 11](#_Toc143531353)

[Endereco\_Pessoa 12](#_Toc143531354)

[Funcao\_Funcionario 13](#_Toc143531355)

[Deposito\_Funcionario 13](#_Toc143531356)

[Servico 14](#_Toc143531357)

[Funcionario 14](#_Toc143531358)

[Tipo\_Hospede 15](#_Toc143531359)

[Hospede 15](#_Toc143531360)

[Tipo\_Quarto 16](#_Toc143531361)

[Quarto 16](#_Toc143531362)

[Tipo\_Pagamento 17](#_Toc143531363)

[Tipo\_Cancelamento 17](#_Toc143531364)

[Tipo\_Contato 17](#_Toc143531365)

[Reserva 18](#_Toc143531366)

[Hospede\_Classe 19](#_Toc143531367)

[Reserva\_Servico 19](#_Toc143531368)

[Estadia 20](#_Toc143531369)

[Avaliacao 20](#_Toc143531370)

[Padronização 21](#_Toc143531371)

[Normalização 21](#_Toc143531372)

# Introdução

## Propósito do projeto

Este projeto visa demonstrar, na prática, os conhecimentos e aprendizados adquiridos nos últimos meses. Nele, irei usar conhecimentos como: construção e normalização de um banco de dados, desenvolvimento de *funcions* e *stored procedures*, criação de *views* para análise de dados por meio de dashboards, backup e aprimoramento de performance e segurança.

## Escopo do projeto

Aqui irei levar em conta um cliente (uma empresa) fictício chamado Pousada Céu Azul.

A empresa é nova no mercado de hotelaria, e ela deseja um banco de dados rápido, eficiente, seguro e que forneça insights importantes para a criação e manutenção de estratégias de negócio.

A pousada conta com diferentes tipos de acomodações e classificação de seus clientes, portanto, ela precisa de um banco de dados que informe a quantidade e o gasto por cada tipo de hóspede, podendo assim desenvolver planos de ação para a empresa.

# Descrição do Cliente

## Resumo da empresa

A construção da empresa começou em 2022, e a finalização da obra está planejada para Junho de 2024. Ela está localizada no Setor Hoteleiro Norte, na Asa Norte de Brasília - DF.

## Hóspedes e Acomodações

A pousada conta com 30 quartos (divididos em: 8 de solteiro duplo, 10 de casais, 7 dormitórios, 2 para famílias e 3 executivos). Além disso, conta com serviços de academia, bar, office room e fornecimento de camas extras.

Os hóspedes são divididos em três categorias: desacompanhado (quando viajam sozinhos), convencional (grupos familiares ou de amigos) e executivo (quando viajam a trabalho). Já as formas de hospedagem no hotel são: por meio do site da empresa, por meio de um dos dois sites terceiros de hospedagem, por telefone e pessoalmente.

# Levantamento de Requisitos

A Pousada deseja um sistema onde ela possa ter um controle sobre quais são os tipos de clientes mais frequentes, os quartos mais utilizados, o tempo de estadia, os motivos de cancelamento de reserva, as avaliações, as principais formas de reserva, os serviços mais utilizados e, caso haja, as várias hospedagens de um mesmo cliente.

## Regras de negócio

Após longas conversas com o cliente, o analista de requisitos chegou nas seguintes regras:

* Cada funcionário só pode exercer uma função e possuir um deposito bancário.
* Apenas os funcionários “Recepcionista” e “Gerente” podem realizar uma reserva.
* Uma reserva pode solicitar nenhum ou vários serviços.
* Academia e Office Room só são pagos uma vez (o cliente pode usar quantas vezes desejar).
* A reserva deve informar: o tipo de pagamento, a forma de reserva, o motivo do cancelamento, quarto reservado e funcionários responsáveis pela reserva.
* A reserva deve ser feita por apenas um hóspede, sendo este um dos hóspedes da estadia.
* Cada reserva não cancelada pode ter várias avaliações, desde que pertençam a tipos diferentes.
* Cada avaliação levará em conta localização, comodidade, limpeza, custo-benefício, conforto e tratamento dos funcionários, pontuando de 0 a 10.
* Cada quarto possui apenas uma categoria, podendo não ter nenhuma em épocas de reforma.
* Cada estadia pode conter vários hóspede, desde que um deles seja o responsável pela reserva.

## Lista de entidades

* Avaliacao.
* Deposito\_funcionario.
* Endereco\_pessoa.
* Funcao\_funionario.
* Funcionario.
* Hospede.
* Quarto.
* Reserva.
* Servico.
* Telefone\_pessoa.
* Tipo\_cancelamento.
* Tipo\_contato.
* Tipo\_hospede.
* Tipo\_quarto.
* Tipo\_pagamento.

### 

### Associativas

* Estadia.
* Hospede\_classe.
* Reserva\_servico.

## Lista de atributos

* Avaliacao: tipo\_contato, id\_reserva, localização, comodidade, limpeza, custo\_benefício, conforto, funcionários.
* Deposito\_funcionario: id\_deposito, codigo\_agencia, numero\_conta, tipo\_conta, nome\_banco.
* Endereco\_pessoa; id\_endereco, país, estado, cidade, bairro, rua, número, complemento, cep, observacao.
* Funcao\_funionario: id\_funcao, nome\_funcao.
* Funcionario: id\_funcionario, nome, sobrenome, dt\_nascimento, cpf, rg, sexo, email, id\_telefone, id\_endereco, tipo\_funcao, conta\_deposito.
* Hospede: id\_hospede, nome, sobrenome, dt\_nascimento, cpf, rg, sexo, email, id\_telefone, id\_endereco.
* Quarto: id\_quarto, andar, tipo\_quarto, camas, valor\_diaria, status.
* Reserva: id\_reserva, data\_checkin, data\_checkout, funcionario\_checkin, funcionario\_checkout, hospede\_principal, numero\_pessoas, id\_quarto, tipo\_pagamento, tipo\_contato, valor\_total, status, cancelamento.
* Servico: id\_servico, nome.
* Telefone\_pessoa: id\_telefone, codigo\_pais, codigo\_area, telefone, observacao.
* Tipo\_cancelamento: id\_tp\_cancelamento, descricao.
* Tipo\_contato: id\_tp\_contato, descricao.
* Tipo\_hospede: id\_tp\_hospede, descricao.
* Tipo\_quarto: id\_tp\_quarto, descricao.
* Tipo\_pagamento: id\_tp\_pagamento, descricao.

Associativas

* Estadia: id\_hospede, id\_reserva.
* Hospede\_classe: id\_tp\_hospede, id\_hospede.
* Reserva\_servico: id\_reserva, id\_servico.

Arquitetura do Banco

# Modelo Conceitual

Diagrama

Descrição gerada automaticamente

# Modelo Lógico

Diagrama

Descrição gerada automaticamente

FAZER UM LÓGICO PARA CADA TABELA

Estruturas do Banco e Tabelas

# Banco de Dados

O banco de dados central da Pousada receberá o nome “db\_ceuazul”. Inicialmente, ela possuirá as seguintes configurações:

|  |  |
| --- | --- |
| MDF size | 102400KB |
| MDF autogrowth | 65536KB |
| LDF size | 25600KB |
| LDF autogrowth | 51200KB |
| Collate | Latin1\_General\_CI\_AS |
| Recovery | Full |
| Page Verify | Checksum |

O cliente deixou claro que poderíamos começar com uma base de dados em um tamanho menor, pois o fluxo de hospedes no começo pode ser baixo. Foi deixado claro que monitoraríamos o crescimento do banco, para a realização de adaptações quando necessário.

A ordenação (collation) foi escolhida com base em pesquisas realizadas pelo próprio cliente. Os dados fornecidos à equipe eram de que mais de 60% dos turistas serão brasileiros, enquanto outros 35% ocidentais, com nomes e dados que serão compatíveis ao banco.

Por uma questão de querer mais segurança no início, foram decididos a implementação de Recovery Full (para o armazenamento de todos os dados possíveis) e Page Verify Checksum (para uma maior integridade dos dados).

# Schemas

Foram criados 5 (cinco) schemas para o banco de dados, a fim de organizar melhor os objetos e garantir mais segurança para o cliente. Os schemas são: Pessoa (contendo dados de telefone e residência de hóspedes e funcionários), Funcionário (contendo os dados de funcionários e suas atividades na empresa), Hóspede (contendo os dados dos hóspedes), Comodo (contendo as tabelas de ‘*quarto’* e ‘*avaliação’*) e Reserva (unindo as principais tabelas sobre dados importantes da cada reserva feita).

A seguir, os schemas criados e as tabelas de cada um:

|  |  |
| --- | --- |
| **Pessoa** | tb\_endereco\_pessoa |
| tb\_telefone\_pessoa |
| **funcionario** | tb\_funcao\_funcionario |
| tb\_funcionario |
| tb\_deposito\_funcionario |
| tb\_servico |
| **hospede** | tb\_estadia |
| tb\_hospede |
| tb\_hospede\_classe |
| tb\_tp\_hospede |
| **comodo** | tb\_avaliacao |
| tb\_tp\_quarto |
| tb\_quarto |
| **reserva** | tb\_reserva |
| tb\_reserva\_servico |
| tb\_tp\_cancelamento |
| tb\_tp\_contato |
| tb\_tp\_pagamento |

# Tabelas

A seguir, cada uma das tabelas do banco de dados de modo detalhado:

## Telefone\_Pessoa

Descrição: Tabela para armazenar dados claros dos telefones de hóspedes e funcionários

Propósito: Armazena dados para fins de contato e geração de insights

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nome da Coluna | Tipo de Dado | Tamanho | Constraint | Aceita NULL | Descrição |
| id\_telefone | INT |  | Primary Key | Não | identificador único de cada telefone |
| codigo\_pais | SMALLINT |  |  | Não | código válido de um país |
| codigo\_area | SMALLINT |  |  | Não | código de área do telefone |
| telefone | INT |  |  | Não | número de telefone da pessoa |
| observacao | VARCHAR | 150 |  | Sim | observações adicionais caso necessário |

Endereco\_Pessoa

Descrição: Tabela para armazenar dados claros dos endereços de hóspedes e funcionários

Propósito: Armazena dados para fins de contato e geração de insights

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_endereco | INT |  | Primary Key | Não | identificador único de cada endereço |
| pais | VARCHAR | 60 |  | Não | nome do país onde a pessoa mora |
| estado | VARCHAR | 60 |  | Não | nome do estado onde a pessoa mora |
| cidade | VARCHAR | 60 |  | Não | nome da cidade onde a pessoa mora |
| bairro | VARCHAR | 60 |  | Não | nome do bairro da cidade |
| rua | VARCHAR | 60 |  | Não | nome da rua onde a pessoa mora |
| numero | INT |  |  | Não | número de residência |
| complemento | VARCHAR | 20 |  | Sim | dados adicionais caso necessários |
| cep | INT |  |  | Não | número de cep da residência |
| observacao | VARCHAR | 150 |  | Sim | observações adicionais caso necessário |

Funcao\_Funcionario

Descrição: Tabela contendo todas as funções realizadas na pousada

Propósito: Armazena as funções dos funcionários

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_funcao | TINYINT |  | Primary Key | Não | identificador único |
| descricao | VARCHAR | 20 |  | Não | nomes das funções prestadas |

Deposito\_Funcionario

Descrição: Tabela para armazenar dados das contas financeiras dos funcionários

Propósito: Armazena dados usados em situações de pagamento de salário

Restrição: Foi utilizado “MASKED WITH” nas colunas “numero\_conta”, “codigo\_agencia” e “tipo\_conta” como segurança básica e inicial das informações sensíveis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_deposito | SMALLINT |  | Primary Key | Não | Identificado único |
| numero\_conta | INT |  |  | Não | número da conta |
| codigo\_agencia | SMALLINT |  |  | Não | código único de cada agência bancária |
| tipo\_conta | TINYINT |  |  | Não | tipo de conta que o funcionário possui |
| nome\_banco | VARCHAR | 15 |  | Não | nome do banco |

Servico

Descrição: Tabela para armazenar os serviços oferecidos pela pousada

Propósito: Armazena dados usados em insights sobre os serviços prestados

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_servico | TINYINT |  | Primary Key | Não | identificador único |
| descricao | VARCHAR | 20 |  | Não | nomes dos serviços oferecidos |
| valor | FLOAT |  |  | Não | valor cobrado pelo serviço |

Funcionario

Descrição: Tabela para armazenar dados pessoais dos funcionários da pousada

Propósito: Armazena detalhes dos funcionários para controle e organização

Restrição: Foi utilizado “MASKED WITH” nas colunas “cpf”, “rg”, “email”, “id\_telefone”, “id\_endereco” e “conta\_deposito” como segurança básica e inicial das informações sensíveis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_funcionario | SMALLINT |  | Primary Key | Não | Identificador único |
| nome | VARCHAR | 20 |  | Não | primeiro nome |
| sobrenome | VARCHAR | 60 |  | Não | sobrenome completo |
| dt\_nascimento | DATE |  |  | Não | data de nascimento |
| cpf | BIGINT |  | Unique | Não | número de cpf |
| rg | INT |  |  | Não | número de rg |
| sexo | CHAR | 1 | Check | Não | m: masculino  f: feminino |
| email | VARCHAR | 50 |  | Não | endereço de email |
| id\_telefone | INT |  | Foreign Key | Não | id do telefone |
| id\_endereco | INT |  | Foreign Key | Não | id do endereço |
| tipo\_funcao | TINYINT |  | Foreign Key | Não | id da função |
| conta\_deposito | SMALLINT |  | Foreign Key | Não | id do depósito |

Tipo\_Hospede

Descrição: Tabela para armazenar as classificações de hóspedes

Propósito: Armazena dados para fins de geração de insights

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_tp\_hospede | TINYINT |  | Primary Key | Não | identificador único |
| descricao | VARCHAR | 20 |  | Não | nomes das categorias de hóspedes |

Hospede

Descrição: Tabela para armazenar dados pessoais dos hóspedes da pousada

Propósito: Armazena detalhes dos hóspedes para controle e organização

Restrição: Foi utilizado “MASKED WITH” nas colunas “cpf”, “rg”, “email”, “id\_telefone” e “id\_endereco” como segurança básica e inicial das informações sensíveis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_funcionario | INT |  | Primary Key | Não | Identificador único |
| nome | VARCHAR | 20 |  | Não | primeiro nome |
| sobrenome | VARCHAR | 60 |  | Não | sobrenome completo |
| dt\_nascimento | DATE |  |  | Não | data de nascimento |
| cpf | BIGINT |  | Unique | Não | número de cpf |
| rg | INT |  |  | Não | número de rg |
| sexo | CHAR | 1 | Check | Não | m: masculino  f: feminino |
| email | VARCHAR | 50 |  | Não | endereço de email |
| id\_telefone | INT |  | Foreign Key | Não | id do telefone |
| id\_endereco | INT |  | Foreign Key | Não | id do endereço |

Tipo\_Quarto

Descrição: Tabela para armazenar classificações de quartos da pousada

Propósito: Armazena dados para fins de controle e geração de insights

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_tp\_quarto | TINYINT |  | Primary Key | Não | identificador único |
| descricao | VARCHAR | 20 |  | Não | nomes das categorias de quartos |

Quarto

Descrição: Tabela para armazenar dados dos quartos disponíveis na pousada

Propósito: Armazena dados para fins de controle e reserva

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_quarto | TINYINT |  | Primary Key | Não | identificador único |
| num\_andar | TINYINT |  |  | Não | número do andar do quarto |
| num\_camas | TINYINT |  |  | Não | número de camas iniciais no quarto |
| valor\_diaria | DECIMAL | (5, 2) |  | Não | valor pago pela diária no quarto |
| tipo\_quarto | TINYINT |  | Foreign Key | Sim | categoria da qual o quarto faz parte |
| status | BIT |  |  | Não | 0 - livre  1 - ocupado |

Tipo\_Pagamento

Descrição: Tabela para armazenar os tipos de pagamentos realizados por hóspedes

Propósito: Armazena dados para fins de controle e geração de insights

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_tp\_pagamento | TINYINT |  | Primary Key | Não | Identificador único |
| descricao | VARHCAR | 20 |  | Não | descreve o tipo de pagamento |

Tipo\_Cancelamento

Descrição: Tabela para armazenar os tipos de cancelamentos realizados por hóspedes

Propósito: Armazena dados para fins de controle e geração de insights

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_tp\_cancelamento | TINYINT |  | Primary Key | Não | Identificador único |
| descricao | VARHCAR | 20 |  | Não | descreve o tipo de cancelamento |

Tipo\_Contato

Descrição: Tabela para armazenar os tipos de contatos realizados por hóspedes

Propósito: Armazena dados para fins de controle e geração de insights

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_tp\_contato | TINYINT |  | Primary Key | Não | Identificador único |
| descricao | VARHCAR | 20 |  | Não | descreve o tipo de contato |

Reserva

Descrição: Tabela para armazenar dados de todas as reservas feitas

Propósito: Armazena dados para gerações de insights

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_reserva | INT |  | Primary Key | Não | identificador único |
| data\_checkin | DATETIME |  |  | Não | data de realização da reserva |
| data\_checkout | DATETIME |  |  | Sim | data de finalização da reserva |
| funcion\_checkin | SMALLINT |  | Foreign Key | Não | funcionário que fez o checkin |
| funcion\_checkout | SMALLINT |  | Foreign Key | Sim | funcionário que fez o checkout |
| hospede\_principal | INT |  | Foreign Key | Não | id do hóspede responsável pelo checkin |
| num\_hospede | TINYINT |  |  | Não | número de hóspedes hospedados na reserva |
| id\_quarto | TINYINT |  | Foreign Key | Não | id do quarto reservado |
| tipo\_pagamento | TINYINT |  | Foreign Key | Não | id do formato de pagamento |
| tipo\_contato | TINYINT |  | Foreign Key | Não | id do formato de contato |
| valor\_total | DECIMAL | (7, 2) |  | Sim | soma do pagamento total a ser realizado |
| status | BIT |  |  | Não | 0 - encerrado  1 - ativo |
| cancelamento | TINYINT |  | Foreign Key | Sim | id do formato de cancelamento, se houver |

Hospede\_Classe

Descrição: Associativa que liga os hóspedes às suas classificações

Propósito: Armazena dados para gerações de insights

Índices: A coluna id\_tp\_hospede será um índice clustered

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_tp\_hospede | TINYINT |  | Primary Key e Foreign Key | Não | id do tipo de hóspede |
| id\_hospede | INT |  | Primary Key e Foreign Key | Não | id do hóspede |

Reserva\_Servico

Descrição: Associativa que liga as reservas aos serviços solicitados

Propósito: Armazena dados para fins de contato e geração de insights

Índices: A coluna id\_reserva será um índice clustered

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_reserva | INT |  | Primary Key e Foreign Key | Não | id da reserva |
| id\_servico | TINYINT |  | Primary Key e Foreign Key | Não | id do serviço solicitado |
| num\_solicitacao | TINYINY |  |  | Não | número de vezes que o serviço foi solicitado |

Estadia

Descrição: Associativa que liga os hóspedes às suas reservas

Propósito: Armazena dados para fins de controle e gerações de insights

Índices: A coluna id\_reserva será um índice clustered

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| id\_hospede | INT |  | Primary Key e Foreign Key | Não | id do hóspede |
| id\_reserva | INT |  | Primary Key e Foreign Key | Não | id da reserva |

Avaliacao

Descrição: Associativa que liga as reservas às suas avaliações

Propósito: Armazena dados para gerações de insights

Índices: A coluna id\_reserva será um índice clustered e a coluna tipo\_contato um índice non clustered

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nome da Coluna** | **Tipo de Dado** | **Tamanho** | **Constraint** | **Aceita NULL** | **Descrição** |
| tipo\_contato | TINYINY |  | Primary Key e Foreign Key | Não | id do tipo de contato |
| id\_reserva | INT |  | Primary Key e Foreign Key | Não | id da reserve |
| comodidade | TINYINT |  | Check | Sim | Nota de 0 a 10 dada pelo hóspede |
| conforto | TINYINT |  | Check | Sim |
| custo\_beneficio | TINYINT |  | Check | Sim |
| localizacao | TINYINT |  | Check | Sim |
| limpeza | TINYINT |  | Check | Sim |

# Padronização

Todas as *CONSTRAINTS* seguiram a mesma forma de criação, sendo escritas detalhadamente no final do *CREATE* para garantir uma padronização em seus nomes.

Os nomes das *CONSTRAINTS* seguiram o seguinte padrão:

tipo.da.constraint\_ + nome.da.coluna\_ + complemento.caso.necessario

Exemplos:

pk\_id\_reserva (Primary Key)

fk\_tp\_contato (Foreign Key)

uk\_cpf\_func (Unique do funcionário)

uk\_cpf\_hosp (Unique do hospede)

Normalização

Antes da criação física de todas as tabelas do projeto, foram realizadas verificações para garantir que os objetos se encontram normalizados.

Primeira Forma Normal: nenhuma tabela apresenta valores que não sejam atômicos ou multivalorados.

Segunda Forma Normal: nenhuma tabela possui atributos que não dependam exclusivamente da chave primária.

Terceira Forma Normal: nenhum tabela contém objetos que sejam transitivamente dependentes de outro objeto.