

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](#_Toc143005727)

[Introdução 4](#_Toc143005728)

[Propósito do projeto 4](#_Toc143005729)

[Escopo do projeto 4](#_Toc143005730)

[Descrição do Cliente 5](#_Toc143005731)

[Resumo da empresa 5](#_Toc143005732)

[Hóspedes e Acomodações 5](#_Toc143005733)

[Levantamento de Requisitos 6](#_Toc143005734)

[Regras de negócio 6](#_Toc143005735)

[Lista de entidades 6](#_Toc143005736)

[Associativas 7](#_Toc143005737)

[Lista de atributos 7](#_Toc143005738)

[Associativas 7](#_Toc143005739)

[Modelo Conceitual 8](#_Toc143005740)

[Modelo Lógico 9](#_Toc143005741)

[Banco de Dados 10](#_Toc143005742)

[Schemas 10](#_Toc143005743)

[Tabelas 11](#_Toc143005744)

[Normalização 16](#_Toc143005745)

# 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 possui 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 e o motivo do cancelamento, quarto reservado e funcionário responsável 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.
* 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\_funcionario |
| tb\_deposito\_funcionario |
| tb\_funcao\_funcionario |
| tb\_servico |
| **hospede** | tb\_hospede |
| tb\_hospede\_classe |
| tb\_tp\_hospede |
| tb\_estadia |
| **comodo** | tb\_quarto |
| tb\_tp\_quarto |
| tb\_avaliacao |
| **reserva** | tb\_reserva |
| tb\_reserva\_servico |
| tb\_tp\_pagamento |
| tb\_tp\_cancelamento |
| tb\_tp\_contato |

# Tabelas

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

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

## 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.

## 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)