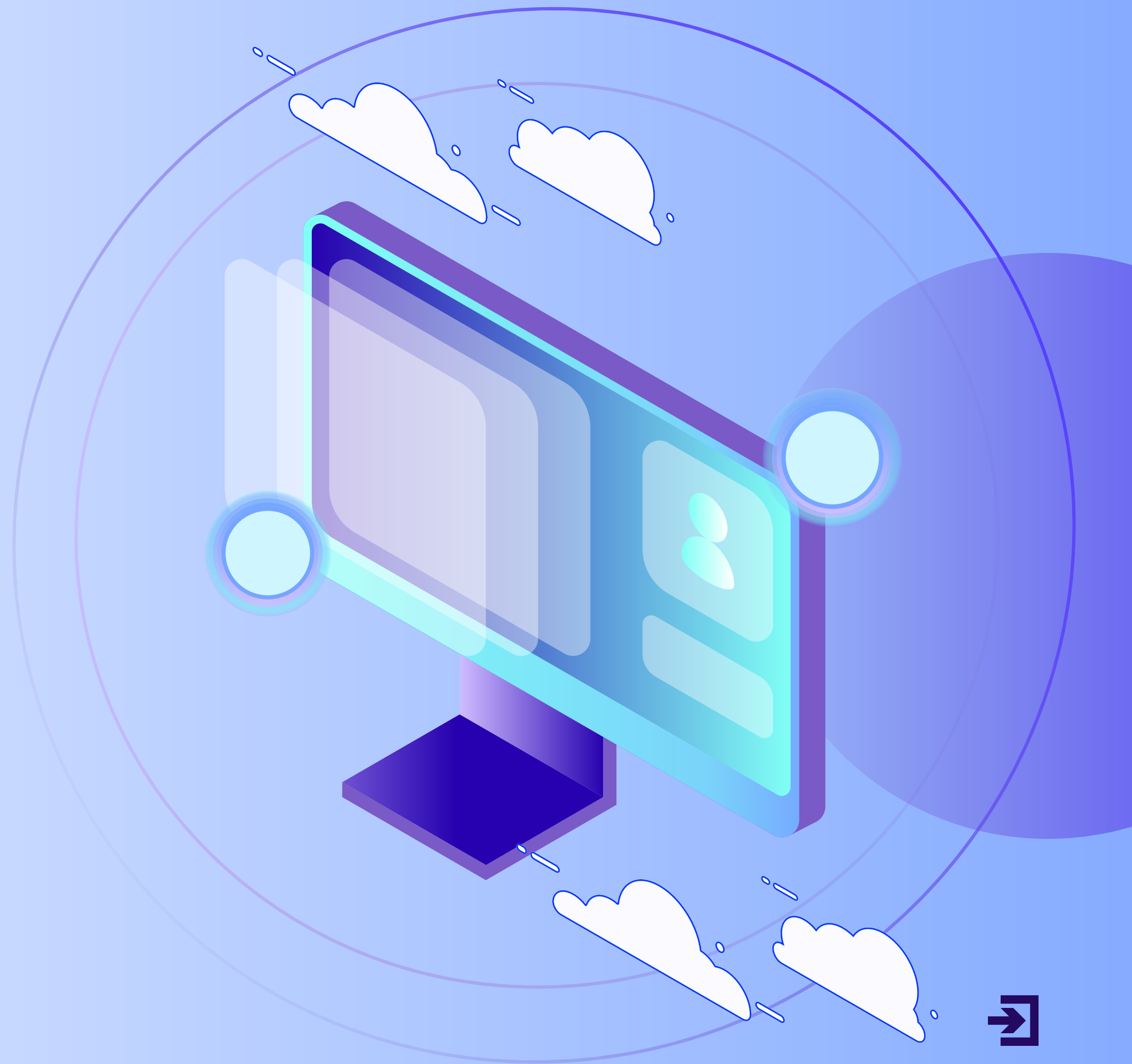




**BANCO DE DADOS**

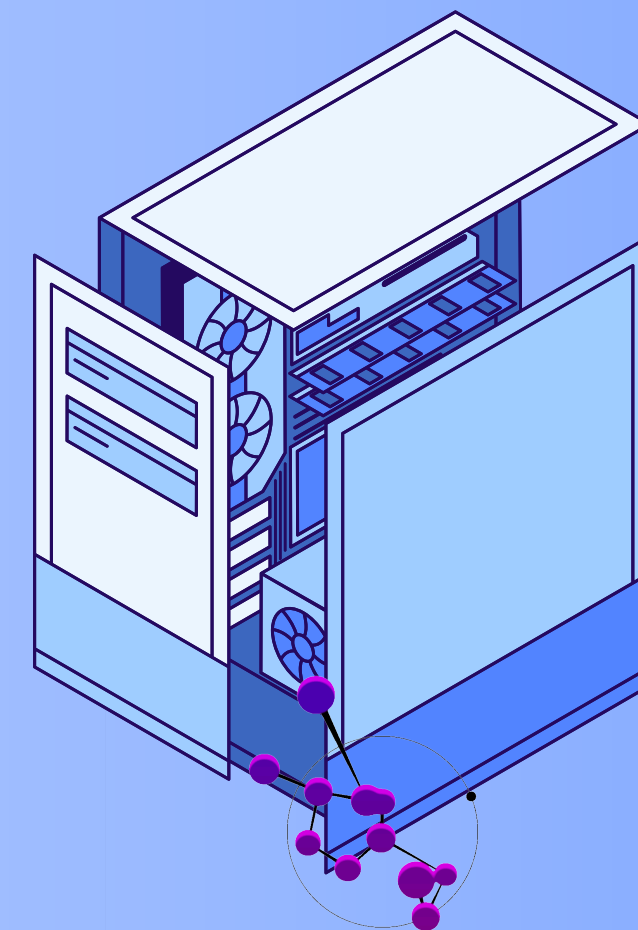
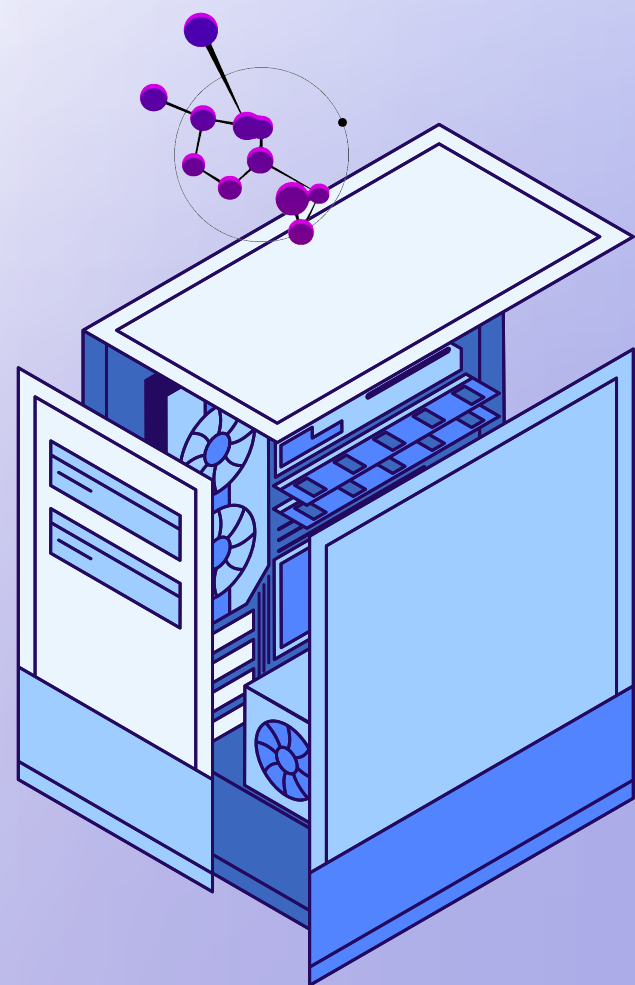
# **HOTEL EASE**

**André Castro Saraiva Leite  
Antônio Roner Ferreira Lô  
Cicero Igor Alves Torquato dos Santos  
Lucas Silva da Costa  
Victor Cleyton de Andrade Chaves**



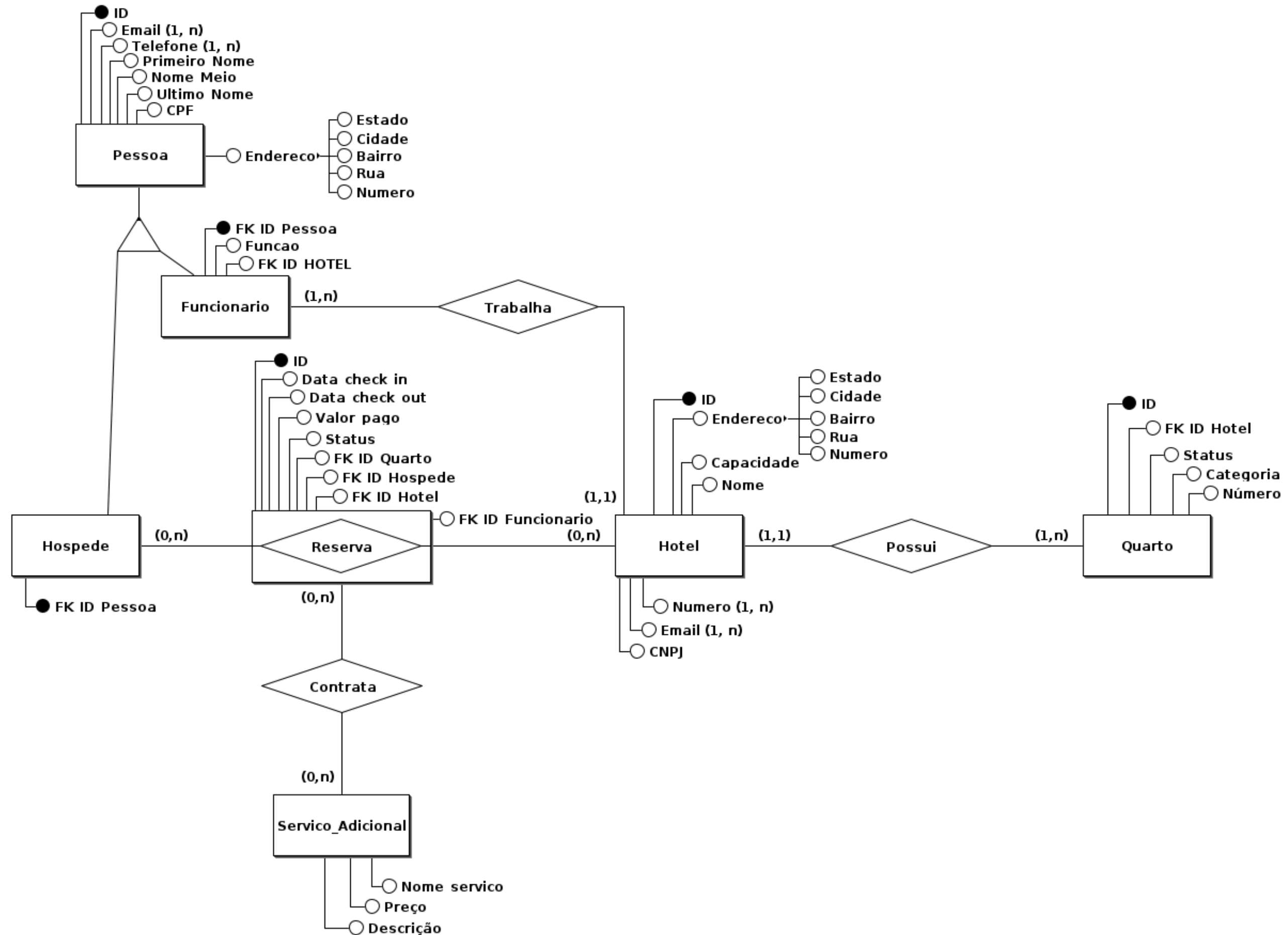
# DESCRIÇÃO

- O **Hotel Ease** é um sistema de gerenciamento hoteleiro projetado para simplificar e automatizar as operações diárias de hotéis. Seu objetivo principal é centralizar informações sobre reservas, hóspedes, quartos, serviços e pagamentos, proporcionando uma experiência mais eficiente tanto para os funcionários quanto para os clientes.





# MODELO ENTIDADE RELACIONAMENTO

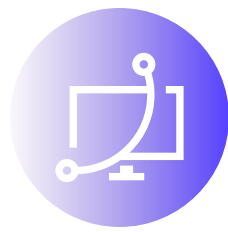




# NORMALIZAÇÃO (1ª FORMA)

- Garantir que cada campo tenha **valores atômicos**
- Remover colunas **repetidas**
- Ter **chave primária** definida

<u>ID</u>	CPF	PrimeiroNome	NomeMeio	UltimoNome
1	000.000.000-01	Carlos		Silva
2	000.000.000-02	Maria	Lima	Souza
3	000.000.000-03	Ana		Ferreira
4	000.000.000-04	João	Pedro	Pereira
5	000.000.000-05	Fernanda		Oliveira



# NORMALIZAÇÃO (2º FORMA)

- Se a tabela tiver **chave primária composta**, garantir que todos os outros campos dependam da chave inteira, não só de parte dela.



```
1 CREATE TABLE Contrata(  
2     FkReserva INT(11) NOT NULL,  
3     FkServicoAdicional INT(11) NOT NULL,  
4     PRIMARY KEY (FkReserva, FkServicoAdicional),  
5     FOREIGN KEY(FkReserva) REFERENCES Reserva(ID),  
6     FOREIGN KEY(FkServicoAdicional) REFERENCES ServicoAdicional(ID)  
7 );
```



# NORMALIZAÇÃO (3<sup>o</sup> FORMA)

- Não deve haver **dependências transitivas**, ou seja, atributos não-chave não podem depender de outro atributo não-chave.

<u>FkPessoa</u>	FkHotel	Funcao
5	1	Recepcionista



# ESTRUTURA DAS TABELAS

```
1 CREATE TABLE Reserva(  
2     ID INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,  
3     DataCheckIn DATE NOT NULL,  
4     DataCheckOut DATE NOT NULL,  
5     ValorPago DOUBLE(8, 2) NOT NULL,  
6     Status VARCHAR(10) NOT NULL CHECK(Status IN('Em andamento', 'Concluída', 'Cancelada')),  
7     FkHotel INT(11) NOT NULL,  
8     FkQuarto INT(11) NOT NULL,  
9     FkHospede INT(11) NOT NULL,  
10    FOREIGN KEY(FkHotel) REFERENCES Hotel(ID),  
11    FOREIGN KEY(FkQuarto) REFERENCES Quarto(ID),  
12    FOREIGN KEY(FkHospede) REFERENCES Hospede(FkPessoa)  
13 );  
14
```



# ESTRUTURA DAS TABELAS

```
1 CREATE TABLE Pessoa(  
2     ID INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,  
3     CPF CHAR(14) UNIQUE,  
4     PrimeiroNome VARCHAR(30) NOT NULL,  
5     NomeMeio VARCHAR(30),  
6     UltimoNome VARCHAR(30)  
7 );  
8  
9 CREATE TABLE PessoaEndereco(  
10     ID INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,  
11     FkPessoa INT(11) NOT NULL,  
12     Rua VARCHAR(30) NOT NULL,  
13     Numero CHAR(8),  
14     Bairro VARCHAR(30) NOT NULL,  
15     Cidade VARCHAR(20) NOT NULL,  
16     Estado VARCHAR(20) NOT NULL,  
17     Cep CHAR(10),  
18     FOREIGN KEY(FkPessoa) REFERENCES Pessoa(ID)  
19 );
```





# CONSULTAS SQL

- 1. Liste o nome, telefone e e-mail de todos os hóspedes cadastrados.
- 2. Exiba as reservas ativas atualmente, com data de check-in e check-out.
- 3. Mostre o total de quartos disponíveis por categoria em um hotel específico.
- 4. Liste as reservas realizadas nos últimos 7 dias, com o nome do hóspede e o valor pago.

```
1  -- 1. Hospedes e seus contatos
2  SELECT
3      Pessoa.PrimeiroNome,
4      Pessoa.UltimoNome,
5      PessoaContato.Telefone,
6      PessoaContato.Email
7  FROM Pessoa
8  JOIN PessoaContato ON PessoaContato.FkPessoa = Pessoa.ID
9  WHERE Pessoa.ID IN (
10     SELECT FkPessoa FROM Hospede
11 );
12
13 -- 2. Reservas em andamento
14 SELECT
15     DataCheckIn,
16     DataCheckOut
17 FROM Reserva
18 WHERE Status = "Em andamento";
19
20 -- 3. Quartos disponíveis por categoria em um hotel
21 SELECT
22     Categoria,
23     COUNT(*)
24 FROM Quarto
25 WHERE Status = "Disponível"
26     AND FkHotel = -ID do hotel-
27 GROUP BY Categoria;
28
29 -- 4. Hospedes com reservas a partir de 7 dias atrás
30 SELECT
31     Pessoa.PrimeiroNome,
32     Pessoa.UltimoNome,
33     Reserva.ValorPago
34 FROM Pessoa
35 JOIN Reserva ON Reserva.FkHospede = Pessoa.ID
36 WHERE Reserva.DataCheckIn >= "Data de hoje com 7 dias a menos";
```



# CONSULTAS SQL

- 5. Exiba os hóspedes que realizaram mais de 3 reservas nos últimos 12 meses.
- 6. Mostre a taxa de ocupação de cada hotel no último mês.
- 7. Liste os hotéis que possuem mais de 90% de ocupação atualmente.
- 8. Exiba os serviços adicionais contratados pelos hóspedes nas últimas 24 horas.

```
1  -- 5. Pessoas com mais de 3 reservas desde o mesmo dia do ano passado
2  SELECT
3      PrimeiroNome,
4      UltimoNome
5  FROM Pessoa
6  JOIN Reserva ON Pessoa.ID = Reserva.ID
7  WHERE Reserva.DataCheckIn >= "Data de hoje no ano passado"
8  GROUP BY Pessoa.ID
9  HAVING COUNT(Reserva.ID) > 3;
10
11 -- 6. Número de reservas por hotel desde o mês passado
12 SELECT
13     Hotel.Nome,
14     COUNT(Reserva.ID)
15 FROM Hotel
16 JOIN Reserva ON Reserva.FkHotel = Hotel.ID
17 WHERE Reserva.DataCheckIn >= "Data de hoje no mês passado"
18 GROUP BY Hotel.ID;
19
20 -- 7. Hotéis com mais de 90% de ocupação
21 SELECT
22     Hotel.Nome
23 FROM Hotel
24 JOIN Quarto ON Quarto.FkHotel = Hotel.ID
25 GROUP BY Hotel.ID, Hotel.Capacidade
26 HAVING (
27     COUNT(CASE WHEN Quarto.Status = "Ocupado" THEN 1 END) / Hotel.Capacidade
28 ) > 0.9;
29
30 -- 8. Serviços adicionais contratados para reservas recentes
31 SELECT
32     ServicoAdicional.*
33 FROM ServicoAdicional
34 JOIN Contrata ON Contrata.FkServicoAdicional = ServicoAdicional.ID
35 WHERE Contrata.FkReserva IN (
36     SELECT Reserva.ID
37     FROM Reserva
38     WHERE DataCheckIn >= "2025-05-09"
39 );
```



# CONSULTAS SQL

- 9. Mostre o faturamento total de cada hotel nos últimos 6 meses.
- 10. Liste os hóspedes que nunca realizaram cancelamentos de reservas.
- 11. Liste os nomes dos hóspedes e os hotéis onde realizaram reservas, com as respectivas datas de check-in e check-out.
- 12. Mostre o total de serviços adicionais contratados por hotel.

```
1  -- 9. Receita por hotel incluindo serviços adicionais (últimos 6 meses)
2  SELECT
3      Hotel.Nome,
4      (SUM(Reserva.ValorPago) + SUM(ServicoAdicional.Preco))
5  FROM Hotel
6  JOIN Reserva ON Reserva.FkHotel = Hotel.ID
7  LEFT JOIN Contrata ON Contrata.FkReserva = Reserva.ID
8  LEFT JOIN ServicoAdicional ON ServicoAdicional.ID = Contrata.FkServicoAdicional
9  WHERE Reserva.DataCheckIn >= "Data dos últimos 6 meses"
10 GROUP BY Hotel.ID;
11
12 -- 10. Pessoas que nunca cancelaram reservas
13 SELECT
14     Pessoa.PrimeiroNome,
15     Pessoa.UltimoNome
16 FROM Pessoa
17 JOIN Hospede ON Pessoa.ID = Hospede.FkPessoa
18 WHERE Hospede.FkPessoa NOT IN (
19     SELECT FkHospede
20     FROM Reserva
21     WHERE Status = "Cancelada"
22 );
23
24 -- 11. Dados de reservas com hóspede e hotel
25 SELECT
26     Pessoa.PrimeiroNome,
27     Pessoa.UltimoNome,
28     Hotel.Nome,
29     Reserva.DataCheckIn,
30     Reserva.DataCheckOut
31 FROM Reserva
32 JOIN Hotel ON Hotel.ID = Reserva.FkHotel
33 LEFT JOIN Pessoa ON Reserva.FkHospede = Pessoa.ID;
34
35 -- 12. Quantidade de serviços adicionais por hotel
36 SELECT
37     Hotel.Nome,
38     COUNT(ServicoAdicional.ID)
39 FROM Hotel
40 JOIN Reserva ON Reserva.FkHotel = Hotel.ID
41 LEFT JOIN Contrata ON Contrata.FkReserva = Reserva.ID
42 LEFT JOIN ServicoAdicional ON ServicoAdicional.ID = Contrata.FkServicoAdicional
43 GROUP BY Hotel.ID;
```



BANCO DE DADOS



# DEMONSTRAÇÃO





# DESAFIOS E APRENDIZADOS



## PRINCIPAIS DESAFIOS

1. Modelagem de dados complexa
2. Gestão de transações em tempo real
3. Segurança e conformidade



## SOLUÇÕES APLICADAS

1. Modelagem de dados dinâmica
2. Otimização de disponibilidade e cancelamentos
3. Criptografia e restrição de dados a pessoal autorizado



## LIÇÕES APRENDIDAS

1. Maximizar a eficiência em gestão
2. Capacidade organizacional



**FIM**

**OBRIGADO**