

1. INTRODUÇÃO

O software será projetado com o intuito de auxiliar em todo o processo da administração de uma pousada. Englobando procedimentos para auxiliar funcionários a efetuarem as reservas de quartos, devoluções, reclamações, manutenção de estoque, tarefas de funcionários, cardápio, entre outras funções que podem ser solicitadas pela gerência posteriormente.

2. REQUISITOS DO PROJETO

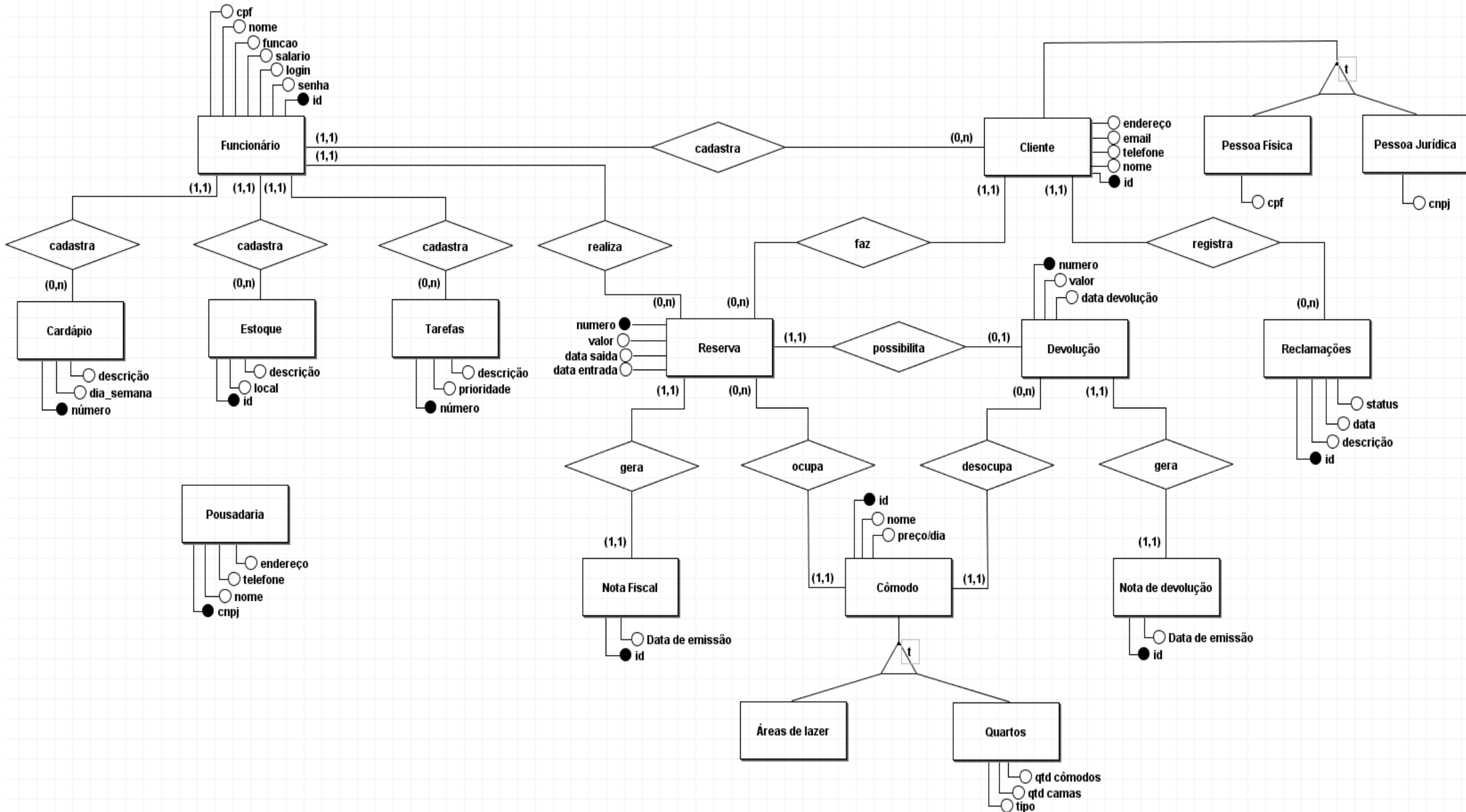
Requisitos do sistema
<ol style="list-style-type: none">1. Manter cadastro de funcionários.2. Manter cadastro de clientes.3. Permitir a consulta de quartos vagos.4. Registrar a reserva de quartos e/ou áreas de lazer.5. Gerar fatura da reserva para o cliente.6. Registrar a devolução do quarto.7. Registrar uma avaliação, reclamação ou sugestão de clientes.8. Gerenciar cardápio.9. Gerenciar estoque de produtos.10. Gerar lista de tarefas.

Regras de Negócio

1. Os clientes da pousada **não poderão** fazer login no sistema.
2. Somente **funcionários cadastrados** podem logar e administrar todas as funcionalidades do sistema.
3. Funcionários são responsáveis por **manter o cadastro de clientes**.
4. Apenas quartos e áreas de lazer **disponíveis** podem ser reservados.
5. Quartos e áreas de lazer podem ser reservados **por até um cliente**.
6. Quartos e áreas de lazer só podem ser reservados para **data atual ou futura**.
7. Apenas quartos e áreas de lazer **reservados** podem ser devolvidos.
8. **Apenas clientes** podem fazer reclamações/sugestões (por intermédio de um funcionário).
9. **Emitir nota fiscal** após a reserva (uma nota fiscal por reserva).
10. **Emitir comprovante de devolução** após a devolução (um comprovante por devolução).

3. MODELAGEM RELACIONAL

3.1 Diagrama Entidade Relacionamento



3.2 Modelagem lógica

Funcionário(id, senha, login, salario, funcao, nome, cpf)
login, cpf devem ser únicos
salario > 0
todos os valores devem ser não-nulos

Cardápio(numero, descricao, dia_semana)
dia_semana tem valor único

Estoque(id, descricao, local)
local tem valor único

Tarefas(numero, descricao, prioridade)
prioridade tem valor único

Cliente(id, endereco, email, telefone, nome, cadastrado_por, tipo)
cadastrado_por referencia Funcionário
tipo define se será PessoaFisica ou PessoaJuridica
todos os valores devem ser não-nulos

PessoaFisica(cpf)
PessoaJuridica(cnpj)

Reserva(numero, valor, dataSaida, dataEntrada, cadastrado_por, reservado_por, idComodo)
cadastrado_por referencia Funcionário
reservado_por referencia Cliente
idComodo referencia comodo
valor > 0
dataSaida >= dataEntrada
todos os atributos devem ser não-nulos

NotaFiscal(id, dataEmissao, numeroReserva)
numeroReserva referencia Reserva

Devolução(numero, valor, dataDevolucao, numeroReserva)
numeroReserva referenciam Reserva
valor=valorReserva se não for atribuído
todos os atributos devem ser não-nulos

NotaDevolucao(id, dataEmissao, numeroDevolucao)
numeroDevolucao referencia Devolução

Comodo(id_comodo, nome, preco_dia, tipoComodo)
tipoComodo define se é área de lazer ou quarto
preco_dia > 0

AreadeLazer()

Quartos(qtdComodos, qtdCamas, tipo)
tipo define se o quarto é suíte, chalé ou solteiro

Reclamações(id, descricao, data, status, id_cliente)
id_cliente referencia cliente
status define se a reclamacao está em aberto ou resolvida

Pousadaria(cnpj, nome, telefone, endereco)
todos os atributos devem ser não-nulos

4. CONSULTAS EM ÁLGEBRA RELACIONAL

Consulta 1: Exiba o id_func, nome, funcao e salario dos funcionários que têm salário maior que R\$1000.

π id_func, nome, funcao, salario σ (salario>1000) Funcionario

|||

Consulta 2: Retorne o nome e telefone dos clientes que são Pessoa Física.

π nome, telefone σ (tipo='Pessoa Física') Cliente

|||

Consulta 3: Retorne o nome do funcionário que cadastrou o cliente 'Faustão'.

π nome σ id_func=(π cadastrado_por σ Cliente.nome='Faustão' Cliente) Funcionario

|||

Consulta 4: Consultar dia_semana e descricao do cardápio para o final de semana.

π dia_semana, descricao σ (dia_semana='Sabado' OR dia_semana='Domingo') Cardapio

|||

Consulta 5: Exiba todos os dados do estoque que não sejam para o local 'Recepção'.

π id, descricao, local, cadastrado_por σ (local<>'Recepção') Estoque

|||

Consulta 6: Busque todas as tarefas com prioridade 'media'.

π numero_tarefa, descricao, prioridade, cadastrado_por σ (prioridade='media') Tarefas

|||

Consulta 7: Consulte todas as reclamações feitas pelo cliente “Elon-musk”.

π descricao, status σ id_cliente=(π id_cliente σ (nome='Elon-musk') Cliente) Reclamacoes

|||

Consulta 8: Consulte todas as reclamações “Em aberto”.

π descricao, status σ (status='Em aberto') Reclamacoes

|||

Consulta 9: Exiba os cômodos que tenham diária menor ou igual a R\$100.

π tipoComodo, nome, preco_dia σ (preco_dia<=100) Comodo

|||

Consulta 10: Exiba o nome de todas as áreas de lazer.

π nome σ (tipoComodo='Area de Lazer') Comodo

|||

Consulta 11: Exiba o número das reservas e os valores das reservas da cliente 'Priscila Alcantara'.

π numero, valor σ reservado_por=(π id_cliente σ (Cliente.nome='Priscila Alcantara') Cliente) Reserva

|||

Consulta 12: Busque o nome e tipo de cômodo reservado pelo cliente 'Elon-musk'.

π nome, tipoComodo σ (id_comodo = (π id_comodo σ reservado_por=(π id_cliente σ (Cliente.nome = 'Elon-musk') Cliente) Reserva)) Comodo

|||

Consulta 13: Busque as reservas futuras (Depois do dia 04/01/2022).

π * σ dataEntrada >= '2022/01/04' Reserva

|||

Consulta 14: Retorne a data de todas as devoluções e seus respectivos valores.

π dataDevolucao, valor Devolucao

|||

Consulta 15: Consulte as datas das devoluções realizadas pelo cliente 'Yudi Tamagoshi'.

π dataDevolucao σ numeroReserva=(π numero σ reservado_por=(π id_cliente σ (Cliente.nome = 'Yudi Tamagoshi') Cliente) Reserva) Devolucao

|||

Consulta 16: Consulte as datas de devolução da 'Churrasqueira'.

π dataDevolucao σ numeroReserva=(π numero σ idComodo=(π id_comodo σ (Comodo.nome = 'Churrasqueira') Comodo) Reserva) Devolucao

5. MODELO FÍSICO (Parte 1) E CARGA DE DADOS

5.1 Tabelas

FUNCIONARIO

```
CREATE TABLE IF NOT EXISTS Funcionario (  
  id_func SERIAL primary key,  
  nome varchar(20) not null,  
  cpf varchar,  
  funcao varchar,  
  salario numeric not null check(salario>0),  
  login varchar(20) not null,  
  senha varchar(10) not null,  
  UNIQUE(login)  
);
```

|||

CLIENTE

```
CREATE TABLE IF NOT EXISTS Cliente (  
  id_cliente SERIAL primary key,  
  nome varchar(30) not null,  
  telefone varchar(30),  
  email varchar,  
  endereco varchar,  
  cpf_cnpj varchar UNIQUE,  
  cadastrado_por INTEGER not null REFERENCES Funcionario(id_func)  
);
```

|||

CARDAPIO


```
CREATE TABLE IF NOT EXISTS Cardapio (  
    numero SERIAL primary key,  
    dia_semana varchar(20) not null,  
    descricao varchar,  
    cadastrado_por INTEGER not null REFERENCES Funcionario(id_func),  
    UNIQUE(dia_semana)  
);
```

|||

ESTOQUE

```
CREATE TABLE IF NOT EXISTS Estoque (  
    id SERIAL primary key,  
    descricao varchar(20) not null,  
    local varchar,  
    cadastrado_por INTEGER not null REFERENCES Funcionario(id_func)  
);
```

|||

TAREFAS

```
CREATE TABLE IF NOT EXISTS Tarefas (  
    numero_tarefa SERIAL primary key,  
    descricao varchar(100) not null,  
    prioridade varchar,  
    cadastrado_por INTEGER not null REFERENCES Funcionario(id_func)  
);
```

|||

RECLAMACOES

```
CREATE TABLE IF NOT EXISTS Reclamacoes (  
    idrec SERIAL PRIMARY KEY,  
    id_cliente integer References Cliente(id_cliente),  
    descricao varchar,  
    data date,  
    status varchar  
);
```

|||

COMODO

```
CREATE TABLE IF NOT EXISTS Comodo (  
    id_comodo SERIAL primary key,  
    nome varchar,  
    preco_dia numeric check(preco_dia>0),  
    tipoComodo varchar  
);
```

|||

RESERVA

```
CREATE TABLE IF NOT EXISTS Reserva (  
    numero SERIAL primary key,  
    valor numeric check(valor>0),  
    dataEntrada date,  
    dataSaida date CHECK(dataSaida>dataEntrada),  
    cadastrado_por INTEGER REFERENCES Funcionario(id_func),  
    reservado_por INTEGER REFERENCES Cliente(id_cliente),  
    idComodo INTEGER REFERENCES Comodo(id_comodo)  
);
```

|||

DEVOLUCAO

```
CREATE TABLE IF NOT EXISTS Devolucao (  
    numero SERIAL primary key,  
    valor numeric check(valor>0),  
    dataDevolucao date,  
    numeroReserva INTEGER REFERENCES Reserva(numero)  
);
```

NOTAFISCAL

```
CREATE TABLE IF NOT EXISTS NotaFiscal (  
    id INTEGER PRIMARY KEY REFERENCES Reserva(numero),  
    data_emissao date  
)
```

NOTADEVOLUCAO

```
CREATE TABLE IF NOT EXISTS NotaDevolucao (  
  id INTEGER PRIMARY KEY REFERENCES Reserva(numero),  
  data_emissao date)
```

5.2 Verificação

Baseamos nossas verificações em negativas, ou seja, as consultas que devem dar errado, vão dar errado. Assim, mostramos os respectivos erros em vermelho, logo abaixo das operações.

FUNCIONARIO

```
INSERT INTO Funcionario (nome, cpf, funcao, salario, login, senha) VALUES ('Sebastiao Maia', '12345678910', 'admin', -1050.12, 'admin', 'admin');
```

-- Schema Error: error: new row for relation "funcionario" violates check constraint "funcionario_salario_check"

|||

CLIENTE

```
INSERT INTO Funcionario (nome, cpf, funcao, salario, login, senha) VALUES ('Sebastiao Maia', '12345678910', 'admin', 1050.12, 'admin', 'admin');
```

```
INSERT INTO Cliente (nome, cpf_cnpj, telefone, email, tipo, endereco, cadastrado_por) VALUES ('Yudi Tamagoshi', '000.000.000-00', '4002-8922', 'yudiplaystation@sbt.com.br', 'Pessoa Física', 'Rodovia Anhanguera, Km 19, Industrial Anhanguera Av. das Comunicações, 4 Osasco - SP', 1);
```

```
UPDATE Cliente set nome=NULL where nome='Yudi Tamagoshi';
```

-- Schema Error: error: null value in column "nome" of relation "cliente" violates not-null constraint

|||

CARDAPIO

```
INSERT INTO Funcionario (nome, cpf, funcao, salario, login, senha) VALUES ('Sebastiao Maia', '12345678910', 'admin', 1050.12, 'admin', 'admin');
```

```
INSERT INTO Cardapio (dia_semana, descricao, cadastrado_por) VALUES ('Domingo', 'Café da Manhã: Vitamina de Banana e bolo de banana\nAlmoço: Feijoada, Arroz, Farofa e Banananada', 1);
```

```
INSERT INTO Cardapio (dia_semana, descricao, cadastrado_por) VALUES ('Domingo', 'Café e pão e coxinha de galinha', 1);
```

```
-- Schema Error: error: duplicate key value violates unique constraint "cardapio_pkey"
```

|||

ESTOQUE

```
INSERT INTO Funcionario (nome, cpf, funcao, salario, login, senha) VALUES ('Sebastiao Maia', '12345678910', 'admin', 1050.12, 'admin', 'admin');
```

```
INSERT INTO Estoque (descricao, local, cadastrado_por) VALUES ('Grampeador', 'Recepção', 1);
```

```
DELETE FROM Funcionario WHERE id_func=1;
```

```
-- Schema Error: error: update or delete on table "funcionario" violates foreign key constraint "estoque_cadastrado_por_fkey" on table "estoque"
```

|||

TAREFAS

```
INSERT INTO Funcionario (nome, cpf, funcao, salario, login, senha) VALUES ('Sebastiao Maia', '12345678910', 'admin', 1050.12, 'admin', 'admin');
```

```
INSERT INTO Tarefas (descricao, prioridade, cadastrado_por) VALUES ('Comprar painéis', 'minima', 1);
```

```
INSERT INTO Tarefas (descricao, prioridade, cadastrado_por) VALUES ('as', 'minima', 2);
```

```
-- Schema Error: error: insert or update on table "tarefas" violates foreign key constraint "tarefas_cadastrado_por_fkey"
```

|||

RECLAMACOES

```
INSERT INTO Funcionario (nome, cpf, funcao, salario, login, senha) VALUES ('Sebastiao Maia', '12345678910', 'admin', 1050.12, 'admin', 'admin');
```

```
INSERT INTO Cliente (nome, cpf_cnpj, telefone, email, tipo, endereco, cadastrado_por) VALUES ('Yudi Tamagoshi', '000.000.000-00', '4002-8922', 'yudiplaystation@sbt.com.br', 'Pessoa Física', 'Rodovia Anhanguera, Km 19, Industrial Anhanguera Av. das Comunicações, 4 Osasco - SP', 1);
```

```
INSERT INTO Reclamacoes (id_cliente, descricao, data, status) VALUES (1, 'Vazamento', '2020-11-03', 'Em aberto');
```

```
UPDATE Reclamacoes set data='20-11-15' where idrec=0;
```

```
-- Schema Error: error: date/time field value out of range: "20-11-15"
```

|||

COMODO

```
INSERT INTO Comodo (nome, preco_dia, tipoComodo) VALUES ('Suíte 01', 150.000, 'Quarto 01');
```

```
UPDATE Comodo set preco_dia=-000.01 where id_comodo=1;
```

```
-- Schema Error: error: new row for relation "comodo" violates check constraint "comodo_preco_dia_check"
```

|||

RESERVA

```
INSERT INTO Funcionario (nome, cpf, funcao, salario, login, senha) VALUES ('Sebastiao Maia', '12345678910', 'admin', 1050.12, 'admin', 'admin');
```

```
INSERT INTO Cliente (nome, cpf_cnpj, telefone, email, tipo, endereco, cadastrado_por) VALUES ('Yudi Tamagoshi', '000.000.000-00', '4002-8922', 'yudiplaystation@sbt.com.br', 'Pessoa Física', 'Rodovia Anhanguera, Km 19, Industrial Anhanguera Av. das Comunicações, 4 Osasco - SP', 1);
```

```
INSERT INTO Comodo (nome, preco_dia, tipoComodo) VALUES ('Suíte 01', 150.000, 'Quarto 01');
```

```
INSERT INTO Reserva (valor, dataEntrada, dataSaida, cadastrado_por, reservado_por, idComodo) VALUES (250, '2022-01-05', '2022-01-01', 1, 1, 1);
```

```
-- Schema Error: error: new row for relation "reserva" violates check constraint "reserva_check"
```

|||

DEVOLUCAO

```
INSERT INTO Devolucao (valor, dataDevolucao, numeroReserva) VALUES (259, '2022-01-04', 1);
```

```
UPDATE Devolucao set numeroReserva=3 where numeroReserva=1;
```

```
-- Schema Error: error: insert or update on table "devolucao" violates foreign key constraint "devolucao_numeroreserva_fkey"
```

5.3 Carga de Dados

(INSERTs para carga dos dados. Inserir uma quantidade razoável de dados para que façam sentido as consultas que serão feitas posteriormente em SQL)

FUNCIONARIO

```
INSERT INTO Funcionario (nome, cpf, funcao, salario, login, senha) VALUES ('Sebastiao Maia', '12345678910', 'admin', '1050.12', 'admin', 'admin');
```

```
INSERT INTO Funcionario (nome, cpf, funcao, salario, login, senha) VALUES ('Caetano Veloso', '12345678912', 'receptionista', '215.51', 'caet', 'veloz');
```

```
INSERT INTO Funcionario (nome, cpf, funcao, salario, login, senha) VALUES ('Zé Ramalho', '12345678913', 'cantor', '4215.51', 'zezin', 'marralho');
```

|||

CLIENTE

```
INSERT INTO Cliente (nome, cpf_cnpj, telefone, email, tipo, endereco, cadastrado_por) VALUES ('Yudi Tamagoshi', '000.000.000-00', '4002-8922', 'yudiplaystation@sbt.com.br', 'Pessoa Física', 'Rodovia Anhanguera, Km 19, Industrial Anhanguera Av. das Comunicações, 4 Osasco - SP', 1);
```

```
INSERT INTO Cliente (nome, cpf_cnpj, telefone, email, tipo, endereco, cadastrado_por) VALUES ('Priscila Alcantara', '111.111.111-11', '4002-8922', 'pribomdia@sbt.com.br', 'Pessoa Física', 'Rodovia Anhanguera, Km 19, Industrial Anhanguera Av. das Comunicações, 4 Osasco - SP', 1);
```

```
INSERT INTO Cliente (nome, cpf_cnpj, telefone, email, tipo, endereco, cadastrado_por) VALUES ('Elon-Musk', '222.222.222-22', '+1(425) 555-0100', 'musketeiro@spacex.com', 'Pessoa Jurídica', 'Los Angeles, em Hawthorne, Califórnia', 2);
```

```
INSERT INTO Cliente (nome, cpf_cnpj, telefone, email, tipo, endereco, cadastrado_por) VALUES ('Faustão', '333.333.333-33', '4003-8000', 'faustosilva@globo.com', 'Pessoa Física', 'Los Angeles, em Hawthorne, Califórnia', 1);
```

|||

CARDAPIO

```
INSERT INTO Cardapio (dia_semana, descricao, cadastrado_por) VALUES ('Domingo', 'Café da Manhã: Vitamina de Banana e bolo de banana\nAlmoço: Feijoadada, Arroz, Farofa e Banananada', 3);
```

```
INSERT INTO Cardapio (dia_semana, descricao, cadastrado_por) VALUES ('Terça-Feira', 'Café da Manhã: Banana com aveia de banana e pão de banana\nAlmoço: Feijoadada, Arroz, Farofa e suco de banana', 2);
```

```
INSERT INTO Cardapio (dia_semana, descricao, cadastrado_por) VALUES ('Sabado', 'Café da Manhã: Banana de Banana e bananada\nAlmoço: Vaca atolada com banana, Arroz, Farofa e limonada', 1);
```

|||

ESTOQUE

```
INSERT INTO Estoque (descricao, local, cadastrado_por) VALUES ('Grampeador', 'Recepção', 1);
```

```
INSERT INTO Estoque (descricao, local, cadastrado_por) VALUES ('Lapiseira', 'Recepção', 2);
```

```
INSERT INTO Estoque (descricao, local, cadastrado_por) VALUES ('Panela', 'Cozinha', 1);
```

```
INSERT INTO Estoque (descricao, local, cadastrado_por) VALUES ('2 Travesseiros', 'Sala', 3);
```

|||

TAREFAS

```
INSERT INTO Tarefas (descricao, prioridade, cadastrado_por) VALUES ('Comprar panelas', 'minima', 1);
```

```
INSERT INTO Tarefas (descricao, prioridade, cadastrado_por) VALUES ('Trocar sofá', 'media',2);
```

```
INSERT INTO Tarefas (descricao, prioridade, cadastrado_por) VALUES ('Reconstruir garagem', 'maxima',2);
```

```
INSERT INTO Tarefas (descricao, prioridade, cadastrado_por) VALUES ('Criar um Jardim', 'minima',1);
```

```
INSERT INTO Tarefas (descricao, prioridade, cadastrado_por) VALUES ('Limpar a piscina', 'media',1);
```

|||

RECLAMACOES

```

INSERT INTO Reclamacoes (id_cliente, descricao, data, status) VALUES (2,'Vazamento da torneira', '2020/11/03', 'Em aberto');

INSERT INTO Reclamacoes (id_cliente, descricao, data, status) VALUES (1,'Barulho', '2020-12-03', 'Resolvido');

INSERT INTO Reclamacoes (id_cliente, descricao, data, status) VALUES (2,'TV Pifou', '2021-03-13', 'Resolvido');

INSERT INTO Reclamacoes (id_cliente, descricao, data, status) VALUES (3,'Luz piscando', '2021-07-19', 'Em aberto');

INSERT INTO Reclamacoes (id_cliente, descricao, data, status) VALUES (4,'Vazamento da torneira', '2021-11-11', 'Em aberto');

```

|||

COMODO

```

INSERT INTO Comodo (nome, preco_dia, tipoComodo) VALUES ('Suíte 01', '150.00', 'Quarto');

INSERT INTO Comodo (nome, preco_dia, tipoComodo) VALUES ('Solteiro 01', '100.00', 'Quarto');

INSERT INTO Comodo (nome, preco_dia, tipoComodo) VALUES ('Chalé 01', '250.00', 'Quarto');

INSERT INTO Comodo (nome, preco_dia, tipoComodo) VALUES ('Churrasqueira', '95.00', 'Area de Lazer');

INSERT INTO Comodo (nome, preco_dia, tipoComodo) VALUES ('Sauna', '87.00', 'Area de Lazer');

```

|||

RESERVA

```

INSERT INTO Reserva (valor, dataEntrada, dataSaida, cadastrado_por, reservado_por, idComodo) VALUES (250, '2022-01-01', '2022-01-05', 1, 1, 1);

INSERT INTO Reserva (valor, dataEntrada, dataSaida, cadastrado_por, reservado_por, idComodo) VALUES (2400, '2021-12-12', '2021-12-26', 2, 2, 2);

INSERT INTO Reserva (valor, dataEntrada, dataSaida, cadastrado_por, reservado_por, idComodo) VALUES (1723, '2021-02-25', '2021-03-19', 3, 3, 3);

INSERT INTO Reserva (valor, dataEntrada, dataSaida, cadastrado_por, reservado_por, idComodo) VALUES (123, '2022-05-01', '2022-06-01', 1, 4, 4);

INSERT INTO Reserva (valor, dataEntrada, dataSaida, cadastrado_por, reservado_por, idComodo) VALUES (579, '2022-07-07', '2022-07-15', 2, 4, 1);

INSERT INTO Reserva (valor, dataEntrada, dataSaida, cadastrado_por, reservado_por, idComodo) VALUES (687, '2020-12-29', '2021-01-15', 3, 3, 4);

```


|||

DEVOLUCAO

```
INSERT INTO Devolucao (valor, dataDevolucao, numeroReserva) VALUES (259, '2022-01-04', 1);
INSERT INTO Devolucao (valor, dataDevolucao, numeroReserva) VALUES (1521, '2021-02-23', 2);
INSERT INTO Devolucao (valor, dataDevolucao, numeroReserva) VALUES (432, '2022-05-08', 3);
INSERT INTO Devolucao (valor, dataDevolucao, numeroReserva) VALUES (725, '2021-01-14', 5);
```

|||

6. CONSULTAS EM SQL

Consulta 1: Buscar cômodos disponíveis em uma data de entrada e saída usando a sintaxe NOT IN.

```
-- Valores de teste
-- data_entrada_input = '2022-02-01'
-- data_saida_input = '2022-02-10'
```

```
SELECT *
FROM Comodo c
WHERE c.id_comodo NOT IN (
    SELECT r.idComodo
    FROM Reserva r
    -- VERIFICA DATAS
    WHERE ((r.dataEntrada > data_entrada_input AND r.dataSaida < data_entrada_input )
        OR (r.dataEntrada > data_saida_input AND r.dataSaida < data_saida_input)
        OR (r.dataEntrada > data_entrada_input AND r.dataSaida < data_saida_input))
    -- RESERVAS NÃO DEVOLVIDAS
    AND r.numero NOT IN (
        SELECT d.numero
```

```
FROM Devolucao d
)
);
```

|||

Consulta 2: Consultar as reservas ativas de determinado cliente (não devolvidas, já iniciadas) com a sintaxe IN.

```
-- Variável de teste
-- comodo_input=4

SELECT *
FROM Comodo c
WHERE c.id_comodo IN (
    SELECT r.numero
    FROM Reserva r
    WHERE
        -- RESERVAS DE UM DETERMINADO CLIENTE
        r.reservado_por = comodo_input
        -- RESERVAS QUE JÁ INICIARAM (USANDO VARIÁVEL DE DATA ATUAL - PostgreSQL)
        AND r.dataEntrada < CURRENT_DATE
        -- RESERVAS NÃO DEVOLVIDAS
        AND r.numero NOT IN (
            SELECT d.numero
            FROM Devolucao d
        )
);
```

|||

Consulta 3: Descrição detalhada das reservas feita por determinado cliente, use a sintaxe de INNER JOIN.

```
-- Variável de teste
-- cliente_id=4

SELECT r.numero, r.valor, r.dataEntrada,
       r.dataSaida, r.cadastrado_por, r.idComodo,
       c.nome, c.preco_dia, c.tipoComodo
FROM Reserva r
INNER JOIN Comodo c
ON r.idComodo = c.id_comodo
```

```

WHERE
-- RESERVAS DE UM DETERMINADO CLIENTE
r.reservado_por = cliente_id
-- RESERVAS QUE JÁ INICIARAM (USANDO VARIÁVEL DE DATA ATUAL - PostgreSQL)
AND r.dataEntrada < CURRENT_DATE
-- RESERVAS NÃO DEVOLVIDAS
AND r.numero NOT IN (
    SELECT d.numero
    FROM Devolucao d
);

```

|||

Consulta 4: Recuperar informações das reservas e suas respectivas devoluções (não excluir reservas sem devolução) usando a sintaxe LEFT JOIN.

```

SELECT r.numero, r.valor AS valor_reserva, r.dataEntrada, r.dataSaida, d.valor AS valor_devolucao,
d.dataDevolucao
FROM Reserva r
LEFT JOIN Devolucao d
    ON r.numero = d.numero;

```

|||

Consulta 5: Idêntica à consulta 4, usando a sintaxe RIGHT JOIN.

```

SELECT r.numero, r.valor AS valor_reserva, r.dataEntrada, r.dataSaida, d.valor AS valor_devolucao,
d.dataDevolucao
FROM Devolucao d
RIGHT JOIN Reserva r
    ON r.numero = d.numero;

```

7 MODELO FÍSICO (Parte 2)

7.1 Visões

(Determinar pelo menos 2 visões para usuários ou sistemas que vão acessar o seu banco de dados. Informar qual o ator que irá acessar a visão e o porquê a visão ajudará esse sistema).

Visão 1: Visualizar dados das reservas (Funcionário).

Objetivo: Permitir que o funcionário recupere informações das reservas e suas respectivas devoluções.

```
CREATE VIEW ReservaDevolucao
AS
SELECT r.numero, r.valor AS valor_reserva, r.dataEntrada, r.dataSaida, d.valor AS valor_devolucao,
d.dataDevolucao
FROM Reserva r
LEFT JOIN Devolucao d
ON r.numero = d.numero;
```

Visão 2: Consultar reservas não devolvidas por cliente (Funcionário).

Objetivo: Viabilizar que o funcionário consulte a descrição detalhada das reservas feita por determinado cliente que ainda não devolveu sua reserva.

```
CREATE VIEW ReservasNaoDevolvidas
AS
SELECT r.numero, r.reservado_por, cli.nome AS nome_cliente, r.valor, r.dataEntrada,
r.dataSaida, r.cadastrado_por, r.idComodo,
c.nome AS nome_comodo, c.preco_dia, c.tipoComodo
FROM Reserva r
INNER JOIN Comodo c
ON r.idComodo = c.id_comodo
INNER JOIN Cliente cli
ON r.reservado_por = cli.id_cliente
WHERE
-- RESERVAS QUE JÁ INICIARAM (USANDO VARIÁVEL DE DATA ATUAL - PostgreSQL)
r.dataEntrada < CURRENT_DATE
-- RESERVAS NÃO DEVOLVIDAS
AND r.numero NOT IN (
SELECT d.numero
FROM Devolucao d
);
```

7.2 Gatilhos e Funções

TRIGGER 1: Atualiza data de saída na Reserva após Devolução

```
CREATE OR REPLACE FUNCTION devolucao_reserva_atualiza_data_saida()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE Reserva
    SET dataSaida = NEW.dataDevolucao
    WHERE numero = NEW.numero;

    RETURN NEW;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER devolucao_reserva_update AFTER INSERT OR UPDATE ON Devolucao
FOR EACH ROW EXECUTE PROCEDURE devolucao_reserva_atualiza_data_saida();
```

TRIGGER 2: Insere dados da Nota Fiscal após inserção de uma reserva

```
CREATE OR REPLACE FUNCTION reserva_create_notafiscal()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO NotaFiscal (id, data_emissao)
    VALUES (NEW.numero, CURRENT_TIMESTAMP);

    RETURN NEW;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER reserva_notafiscal AFTER INSERT ON Reserva
FOR EACH ROW EXECUTE PROCEDURE reserva_create_notafiscal();
```

7.3 Verificação

– Cadastrando primeira reserva (numero = 1)

```
INSERT INTO Reserva (valor, dataEntrada, dataSaida, cadastrado_por, reservado_por, idComodo) VALUES (2400, '2021-12-12', '2021-12-26', 2, 2, 2);
```

– Nova Nota Fiscal inserida

```
SELECT * FROM NotaFiscal WHERE id = 1;
```

– Nova Nota Fiscal inserida

```
SELECT * FROM NotaFiscal WHERE id = 1;
```

– Registrando Devolução

```
INSERT INTO Devolucao (valor, dataDevolucao, numero) VALUES (1521, '2022-02-23', 1);
```

– Data da Reserva alterada

```
SELECT * FROM Reserva WHERE numero = 1;
```

8 OTIMIZAÇÃO DO BANCO

8.1 Índices

Neste projeto os campos `Cliente.cpf_cnpj` e `Funcionario.cpf` podem atuar como índice, visto que são atributos que apresentam a constraint `UNIQUE`. SGBDs como o PostgreSQL configuram atributos `UNIQUE` como índices, porém isso não impossibilita a configuração manual do mesmo.

```
CREATE INDEX cliente_cpf_cnpj ON Cliente (cpf_cnpj);
```

```
CREATE INDEX funcionario_cpf ON Funcionario (cpf);
```

8.2 Verificação dos índices

Os dados a seguir foram coletados após inserção de 10.000 clientes com `cpf_cnpj` aleatórios. Note que o PostgreSQL utiliza o chamado *shared_buffers*, que armazena blocos de tabelas recentemente visitadas. O *seed* utilizado nos testes segue em anexo.

`SELECT * FROM pousadaria.Cliente WHERE cpf_cnpj = '07470729953';` – existe

`SELECT * FROM pousadaria.Cliente WHERE cpf_cnpj = '07470729954';` – não existe

Com UNIQUE/INDEX		
Valor	'07470729953'	(existe)
Tempos (msec)	34	
	34	
	34	
	32	
	34	
	33	
	33	
	35	
	36	
	34	
Média	33,9	
Mediana	34	
Variância	1,21	
Desvio padrão	1,10	

Com UNIQUE/INDEX		
Valor	'07470729954'	(não existe)
Tempos (msec)	121	
	36	
	33	
	33	
	33	
	34	
	33	
	34	
	32	
	34	
Média	42,3	
Mediana	33,5	
Variância	765,79	
Desvio padrão	27,67	

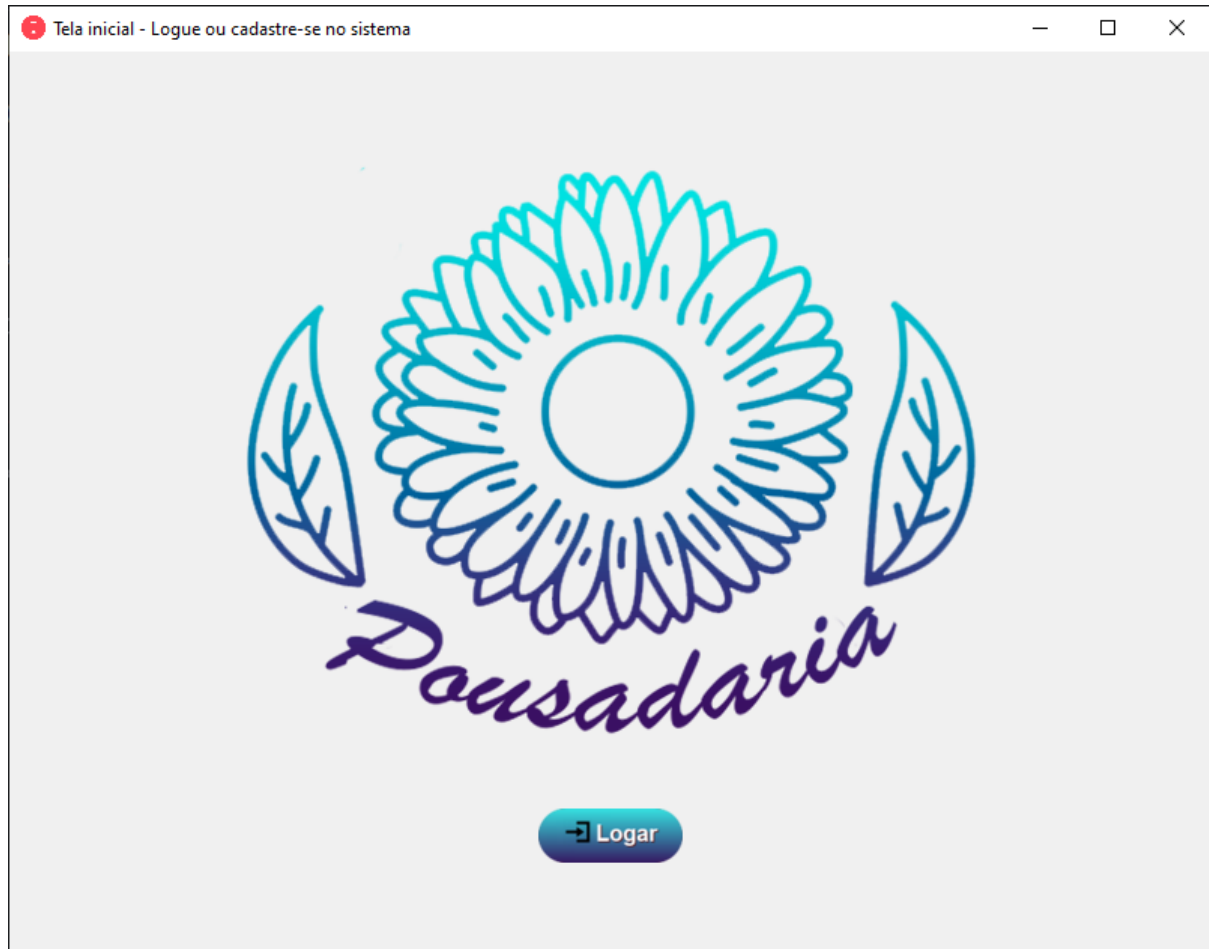
Sem UNIQUE/INDEX		
Valor	'07470729953'	(existe)
Tempos (msec)	39	
	33	
	35	
	37	
	33	
	34	
	33	
	34	
	31	
	33	
Média	34,2	
Mediana	33,5	
Variância	5,3	
Desvio padrão	2,3	

Sem UNIQUE/INDEX		
Valor	'07470729954'	(não existe)
Tempos (msec)	37	
	35	
	32	
	34	
	32	
	35	
	32	
	34	
	33	
	34	
Média	33,8	
Mediana	34	
Variância	2,62	
Desvio padrão	1,62	

9. TELAS DA APLICAÇÃO

Requisitos de negócio

1. Manter cadastro de funcionários.



Início - Login de Funcionário

Insira os dados de Login

Login: Mat

Senha: ***

Usuario Logado! Você já pode usar o sistema!

Voltar

Abrir

Menu Pousadaria - Escolha uma Opção



Recepção

Cadastrar cliente

Quartos Disponíveis

Reservar quartos

Devolução de quartos

Reclamações ou sugestões

Administração - Selecione uma opção

Gerencia cardápio

Controle de estoque

Lista de tarefas

Cadastrar funcionário

Início - Cadastro de funcionários

Insira os dados do Funcionário

Nome:

CPF:

Função:

Recepção

Salario:

Login:

Senha:

Voltar

Cadastrar

Início - Cadastro de funcionários

Insira os dados do Funcionário

Nome:

Mathews

CPF:

44444444444

Função:

Outros

Salario:

2222222222

Login:

Mat


Senha:

Cadastro efetuado com sucesso!

Voltar

2. Manter cadastro de clientes.

Recepção - Cadastro de clientes



Dados do Cliente

Nome:

CPF/CNPJ:

E-mail:

Telefone:

Tipo:

Endereço

Rua:

Numero:

Bairro:

Cidade:

Estado:

[Voltar](#) [Cadastrar](#)

Recepção - Cadastro de clientes



Dados do Cliente

Nome:

CPF/CNPJ:

E-mail:

Telefone:

Tipo:

Endereço

Rua:

Numero:

Bairro:

Cidade:

Estado:

Dados cadastrados com sucesso!

[Voltar](#)

3. Permitir a consulta de quartos vagos.

Recepção - Consulta de Quartos

Filtro de disponibilidade

Status: Disponível

Quartos

====>IDQuarto: 5<====
Nome: Solteiro 02
Preço: 59.27
Tipo: Solteiro
QTD Camas: 2 Cama(s) de solteiro
QTD Cômodos: 2 Cômodos - Quarto e Banheiro

====>IDQuarto: 6<====
Nome: Solteiro 03
Preço: 63.85
Tipo: Solteiro
QTD Camas: 1 Cama(s) de solteiro
QTD Cômodos: 2 Cômodos - Quarto e Banheiro

====>IDQuarto: 7<====
Nome: Chalé 01

Áreas de lazer

====>IDArea: 12 - Salão de Festas<====
Tipo: Área de Lazer
Diária: 125.72

Voltar

Consultar

Recepção - Consulta de Quartos

Filtro de disponibilidade

Status: Ocupado

Quartos

====>IDQuarto: 1<====
Nome: Suite 01
Preço: 150.62
Tipo: Suite
QTD Camas: 1 Cama(s) de casal
QTD Cômodos: 2 Cômodos - Quarto e Banheiro

====>IDQuarto: 2<====
Nome: Suite 02
Preço: 155.15
Tipo: Suite
QTD Camas: 2 Cama(s) de casal
QTD Cômodos: 2 Cômodos - Quarto e Banheiro

====>IDQuarto: 3<====
Nome: Suite 03

Áreas de lazer

====>IDArea: 9 - Churrasqueira<====
Tipo: Área de Lazer
Diária: 95.6

Voltar

Consultar

4. Registrar a reserva de quartos e/ou áreas de lazer.

Recepção - Reservar quartos e/ou áreas de lazer

Selecione um cliente

Cientes:

Yudi Tamagoshi

Data de entrada:

2022-02-19

Data de saída:

2022-02-19

Selecionar

Voltar

ERRO - A data de saída deve ser futura!

Recepção - Reservar quartos e/ou áreas de lazer

Selecione um cliente

Cientes:

Yudi Tamagoshi

Data de entrada:

2022-02-19

Data de saída:

2022-02-20

Selecionar

Selecione os quartos/áreas desejadas

Selecione os quartos desejados

☒ Suite 01

☐ Suite 02

☐ Suite 03

☐ Solteiro 01

☐ Solteiro 02

☐ Solteiro 03

☐ Chalé 01

☐ Chalé 02

Selecione as áreas de lazer desejadas

☒ Churrasqueira

☐ Sauna

☐ Piscina

☐ Salão de Festas

Somatório quartos: 150.62

Somatório áreas: 95.6

Valor da reserva para 1 dias: 246.22

Voltar

Reservar

Recepção - Reservar quartos e/ou áreas de lazer

Selecione um cliente

Clientes: Yudi Tamagoshi

Data de entrada: 2022-02-19

Data de saída: 2022-02-20

Selecionar

Selecione os quartos/áreas desejadas

Selecione os quartos desejados

☒ Suite 01

☐ Suite 02

☐ Suite 03

☐ Solteiro 01

☐ Solteiro 02

☐ Solteiro 03

☐ Chalé 01

☐ Chalé 02

Selecione as áreas de lazer desejadas

☒ Churrasqueira

☐ Sauna

☐ Piscina

☐ Salão de Festas

Somatório quartos: 150.62

Somatório áreas: 95.6

Valor da reserva para 1 dias: 246.22

Voltar

Quartos e dependências reservadas com sucesso!

Emitir Nota

5. Gerar fatura da reserva para o cliente.

Recepção - Reservar quartos e/ou áreas de lazer

Selecione um cliente

Clientes: Yudi Tamagoshi

Data de entrada: 2022-02-19

Data de saída: 2022-02-20

Selecionar

Selecione os quartos/áreas desejadas

Selecione os quartos desejados

☒ Suite 01

☐ Suite 02

☐ Suite 03

☐ Solteiro 01

☐ Solteiro 02

☐ Solteiro 03

☐ Chalé 01

☐ Chalé 02

Selecione as áreas de lazer desejadas

☒ Churrasqueira

☐ Sauna

☐ Piscina

☐ Salão de Festas

Somatório quartos: 150.62

Somatório áreas: 95.6

Valor da reserva para 1 dias: 246.22

Voltar

Quartos e dependências reservadas com sucesso!

Emitir Nota

```
Yudi Tamagoshi.txt
1 #-----Nota Fiscal-----#
2
3 DADOS DO EMITENTE:
4
5 Nome: Pousadaria
6 Telefone: (xx) xxxx-xxxx
7 Endereço: Rua dos bobos, 0
8 CNPJ: xx.xxx.xxx/xxxx-xx
9
10
11 DADOS DO CLIENTE:
12 Nome: Yudi Tamagoshi
13 CPF: 00000000000
14 CNPJ: None
15 Telefone: 4002-8922
16 E-mail: yudiplaystation@sbt.com.br
17 Endereço: Rodovia Anhanguera, Km 19, Industrial Anhanguera Av. das Comunicações, 4 Osasco - SP
18 Quartos Alugados: ['1']
19 Áreas de lazer alugadas: ['9']
20 Tempo de estadia: 1
21 Data de Entrada: 2022-02-19
22 Data de Saída: 2022-02-20
23
24
25 Data de emissão: 2022-02-19          Valor: 246.22
26
27
```

6. Registrar a devolução do quarto.

Recepção - Devoluções

Selecione um cliente

Nomes: Yudi Tamagoshi

Selecionar

Selecione os quartos/areas a serem devolvidos

Selecione os quartos a serem devolvidos

☒ Suite 01

Selecione as áreas de lazer a serem devolvidas

☒ Churrasqueira

Voltar

Devolver

Recepção - Devoluções

Selecione um cliente

Nomes: Yudi Tamagoshi

Selecionar

Selecione os quartos/áreas a serem devolvidos

Selecione os quartos a serem devolvidos

☒ Suite 01

Selecione as áreas de lazer a serem devolvidas

☒ Churrasqueira

Voltar

Nota de devolução emitida com sucesso!

```
Yudi Tamagoshi.txt
1 #-----Nota De Devolução-----
2
3 DADOS DO EMITENTE:
4
5 Nome: Pousadaria
6 Telefone: (xx) xxxx-xxxx
7 Endereço: Rua dos bobos, 0
8 CNPJ: xx.xxx.xxx/xxxx-xx
9
10
11 DADOS DO CLIENTE:
12
13 Nome: Yudi Tamagoshi
14 CPF: 00000000000
15 CNPJ: None
16 Telefone: 4002-8922
17 E-mail: yudiplaystation@sbt.com.br
18 Endereço: Rodovia Anhanguera, Km 19, Industrial Anhanguera Av. das Comunicações, 4 Osasco - SP
19 Quartos Devolvidos: ['17']
20 Áreas de lazer devolvidas: ['18']
21
22
23 Data da devolução: 2022-02-19
24
```

7. Registrar uma avaliação, reclamação ou sugestão de clientes.

Recepção - Reclamações, Sugestões ou Avaliações

Filtragem de funções

Operação:

Inserir

Inserir

Consultar

Atualizar

Deletar

Próximo

Insira a reclamação

Vazamento da torneira

Cliente:

Yudi Tamagoshi

Status:

Resolvida

Inserir

Voltar

Recepção - Reclamações, Sugestões ou Avaliações

Filtragem de funções

Operação:

Consultar

Próximo

Consulta de reclamações

Cliente:

Yudi Tamagoshi

IdReclamação: 5

Cliente: 1

Status: Resolvida

Data e hora: 19/02/2022 (10:40:57)

Mensagem: Vazamento de torneira

=====X=====

Voltar

Processo de consulta realizado com sucesso!

Consultar

Recepção - Reclamações, Sugestões ou Avaliações

Filtragem de funções

Operação: Atualizar

Próximo

Selecione um cliente

Cliente: Yudi Tamagoshi

Consultar

Atualiza reclamação

Status: Em aberto

Vazamento de torneira

Voltar

Reclamação atualizada no banco de dados!

Atualizar

Recepção - Reclamações, Sugestões ou Avaliações

Filtragem de funções

Operação: Deletar

Próximo

Selecione um cliente

Cliente: Yudi Tamagoshi

Deletar

Voltar

Reclamação do cliente deletada do banco de dados com sucesso!

8. Gerenciar cardápio.

Administração - Cardápio

Filtragem de funções

Operação: Inserir

Próximo

Insira a Cardápio

Data: 19/02/2022

Café da manhã:
Mamão papaia, banana, maçã e frutas em geral
Pão integral, pão de forma, pão de sal
Requeijão, Margarina e Manteiga
Leite, café, suco

Voltar

Cardápio registrado!

Inserir

Administração - Cardápio

Filtragem de funções

Operação: Consultar

Próximo

Refeições

Dia: 19/02/2022

Café da manhã:
Mamão papaia, banana, maçã e frutas em geral
Pão integral, pão de forma, pão de sal
Requeijão, Margarina e Manteiga
Leite, café, suco

Voltar

Processo de consulta realizado com sucesso!

Consultar

Administração - Cardápio

Filtragem de funções

Operação: Atualizar

Próximo

Selecione uma data

Cliente: 19/02/2022

Consultar

Edição de Cardápio

Café da manhã:
Frutas, pães e Bebidas

Voltar

Cardápio atualizado no banco de dados!

Atualizar

Administração - Cardápio

Filtragem de funções

Operação: Deletar

Próximo

Selecione uma data

Data: 19/02/2022

Deletar

Voltar

Cardápio deletado do banco de dados com sucesso!

9. Gerenciar estoque de produtos.

Administração - Controle de Estoque

Filtragem de funções

Operação: Inserir

Próximo

Insira o Estoque

Local: Recepção

Guardanapo, panela antiaderente, Arroz, feijão, legumes em geral

Voltar

Estoque registrado!

Inserir

Administração - Controle de Estoque

Filtragem de funções

Operação: Consultar

Próximo

Informações do estoque

Local: Cozinha

Guardanapo, panela antiaderente, Arroz, feijão, legumes em geral

Voltar

Processo de consulta realizado com sucesso!

Consultar

Administração - Controle de Estoque

Filtragem de funções

Operação:

Atualizar

Inserir

Consultar

Atualizar

Deletar

Próximo

Selecione um Local

Local:

Cozinha

Consultar

Edição de Estoque

Guardanapo, panela antiaderente, Arroz, feijão, legumes em geral

Voltar

Estoque atualizado no banco de dados!

Atualizar

Administração - Controle de Estoque

Filtragem de funções

Operação:

Deletar

Próximo

Selecione um Local

Local:

Recepção

Deletar

Voltar

Estoque deletado do banco de dados com sucesso!

10. Gerar lista de tarefas.

Administração - Tarefas

Filtragem de funções

Operação:

Inserir

Inserir

Consultar

Atualizar

Deletar

Próximo

Insira as Tarefas

Prioridade: Urgente

Limpar Piscina

Voltar

Tarefa registrada!

Inserir

Administração - Tarefas

Filtragem de funções

Operação:

Atualizar

Próximo

Selecione um prioridade

Prioridade: Urgente

Consultar

Edição de Tarefas

Limpar Quadra

Voltar

Tarefas atualizadas no banco de dados!

Atualizar

Administração - Tarefas

Filtragem de funções

Operação: Consultar

Próximo

Informações do Tarefas

Prioridade: Urgente

Limpar Piscina

Voltar

Processo de consulta realizado com sucesso!

Consultar

Administração - Tarefas

Filtragem de funções

Operação: Deletar

Próximo

Selecione um prioridade

Prioridade: Urgente

Deletar

Voltar

Tarefas da prioridade selecionada foram deletadas com sucesso!