

Universidade Federal Rural de Pernambuco
Unidade Acadêmica de Serra Talhada
Curso: Sistemas de Informação
Disciplina: Modelagem de Programação Orientada a Objeto

Sistemas de Reserva de Hotel - PART II

Grupo:

Erisvaldo Cleiton de Almeida Lima
Igor Enrique Pereira de Lima
Elielson Vincente de Souza

Professor:

Michael Oliveira da Cruz

Serra Talhada - PE
Setembro, 2024

Sumário

Introdução3

Arquitetura3

Modelos de Dados3

Persistência de Dados3

Endpoints.....4

Imagens e Diagrama5

Conclusão8

Introdução

O projeto Sistema de Registro de Hotel consiste em uma aplicação backend desenvolvida com o framework FastAPI, cujo objetivo é gerenciar operações essenciais em um hotel. O sistema abrange funcionalidades como reservas de quartos, registro de clientes, gerenciamento de itens e compras realizadas no hotel. A aplicação foi projetada para otimizar o controle e a automação de processos administrativos. Inicialmente, a persistência de dados era feita utilizando arquivos JSON, mas agora foi implementado o uso de um banco de dados relacional (SQLite) para maior escalabilidade e integridade dos dados.

Arquitetura

A arquitetura do sistema continua modular, utilizando o FastAPI como framework principal para a criação de APIs. No entanto, a persistência de dados foi aprimorada com a integração do banco de dados SQLite, utilizando SQLAlchemy como ORM para facilitar o mapeamento objeto-relacional. A aplicação está organizada em modelos de dados, rotas para os diferentes endpoints e funções auxiliares responsáveis pelo gerenciamento de sessões com o banco de dados.

Modelos de Dados

Os modelos de dados foram criados com o auxílio das bibliotecas Pydantic e SQLAlchemy. O Pydantic continua a ser utilizado para a validação dos dados nas requisições da API, enquanto o SQLAlchemy permite o mapeamento para o banco de dados SQLite. Os modelos principais incluem:

- Quarto: Representa os quartos disponíveis no hotel, com atributos como ID, número do quarto, tipo e preço.
- Cliente: Contém as informações dos clientes, como ID, nome e CPF.
- Reserva: Registra as reservas feitas, associando clientes a quartos específicos.
- Item: Refere-se aos itens disponíveis para compra dentro do hotel, detalhados com ID, nome e preço do produto.
- Compra: Registra as compras realizadas pelos clientes, ligando cada compra ao respectivo cliente e item adquirido.

Agora, as tabelas e os relacionamentos entre as entidades foram implementados no banco de dados, permitindo consultas mais eficientes e relações consistentes entre clientes, reservas e itens comprados.

Persistência de Dados

Os dados agora são persistidos no banco de dados SQLite, em vez de arquivos JSON. Essa mudança melhora a escalabilidade, integridade e consistência das informações, além de permitir o uso de transações e consultas mais sofisticadas. A migração para SQLite foi implementada sem alterar a estrutura lógica do sistema, mantendo a robustez e otimizando o desempenho.

Endpoints

A API do sistema mantém os mesmos endpoints principais, mas com novas funcionalidades e otimizações para operar com o banco de dados SQLite. Aqui estão alguns dos principais endpoints:

- POST /quartos/: Adiciona novos quartos ao sistema.
- PUT /quartos/{quarto_id}: Edita as informações de um quarto existente.
- DELETE /quartos/{quarto_id}: Remove um quarto.
- POST /clientes/: Registra novos clientes no sistema.
- PUT /clientes/{cliente_id}: Edita as informações de um cliente.
- DELETE /clientes/{cliente_id}: Remove um cliente.
- POST /reservas/: Efetua a reserva de quartos.
- PUT /reservas/{reserva_id}: Edita uma reserva existente.
- DELETE /reservas/{reserva_id}: Cancela uma reserva.
- POST /itens/: Adiciona novos itens ao inventário do hotel.
- POST /compras/: Registra compras realizadas por clientes.
- GET /clientes/{cliente_id}/reservas: Retorna todas as reservas de um cliente específico.
- GET /total_pagar/{cliente_id}: Gera uma nota fiscal detalhada para um cliente, incluindo o total das reservas e compras.

Imagens e Diagrama

Figura 1 - Diagrama de Classe

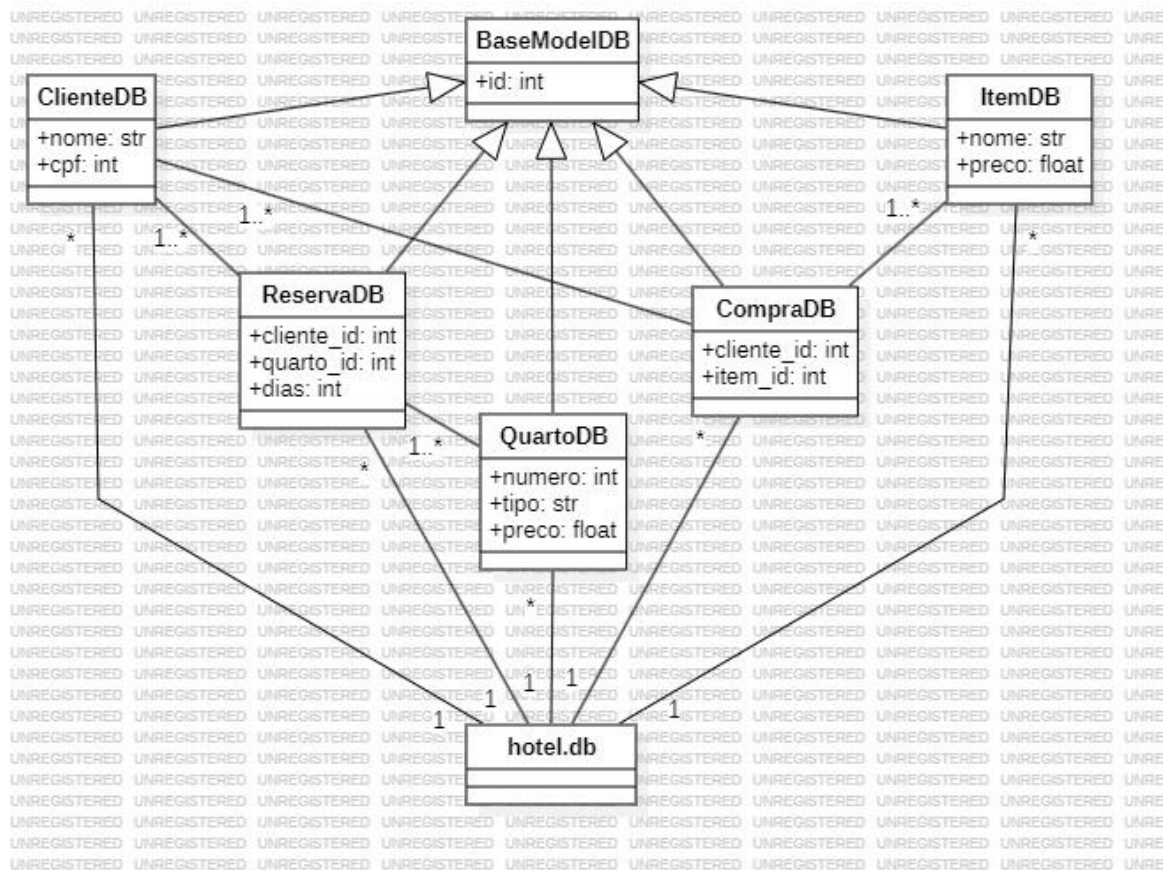


Figura 2 - EndPoints

POST	/quartos/ Adicionar Quarto	✓
PUT	/quartos/{quarto_id} Editar Quarto	✓
DELETE	/quartos/{quarto_id} Remover Quarto	✓
POST	/clientes/ Registrar Cliente	✓
PUT	/clientes/{cliente_id} Editar Cliente	✓
DELETE	/clientes/{cliente_id} Remover Cliente	✓
POST	/reservas/ Fazer Reserva	✓
GET	/total_pagar/{cliente_id} Calcular Nota Fiscal	✓
PUT	/reservas/{reserva_id} Editar Reserva	✓
DELETE	/reservas/{reserva_id} Cancelar Reserva	✓
POST	/itens/ Adicionar Item	✓
PUT	/itens/{item_id} Editar Item	✓
DELETE	/itens/{item_id} Remover Item	✓
GET	/clientes/{cliente_id}/reservas Listar Reservas Por Cliente	✓
POST	/compras/ Comprar Item	✓
GET	/dados_hotel/ Exibir Dados Hotel	✓

Figura 3 - Nota Fiscal do Cliente

Response body

```
{
  "nome_cliente": "Cleiton",
  "reservas": [
    {
      "quarto": "Solteiro",
      "dias": 1,
      "preco_diaria": 95,
      "preco_total": 95
    },
    {
      "quarto": "Suíte",
      "dias": 1,
      "preco_diaria": 180,
      "preco_total": 180
    }
  ],
  "itens_comprados": [
    {
      "nome": "Água",
      "preco": 2.5
    },
    {
      "nome": "Coca",
      "preco": 5
    }
  ],
  "total_reservas": 275,
  "total_compras": 7.5,
  "total_a_pagar": 282.5
}
```

Figura 4 - Todos os dados do Hotel

```
{
  "quartos": [
    {
      "preco": 95,
      "tipo": "Solteiro",
      "id": 1,
      "numero": "01"
    },
    {
      "preco": 120,
      "tipo": "Casal",
      "id": 2,
      "numero": "02"
    },
    {
      "preco": 180,
      "tipo": "Suíte",
      "id": 3,
      "numero": "03"
    }
  ],
  "clientes": [
    {
      "nome": "Cleiton",
      "cpf": 12365478911,
      "id": 1,
      "total_a_pagar": 282.5
    }
  ],
  "reservas": [
    {
      "cliente_id": 1,
      "id": 1,
      "dias": 1,
      "quarto_id": 1
    },
    {
      "cliente_id": 1,
      "id": 3,
      "dias": 1,
      "quarto_id": 3
    }
  ],
  "itens": [
    {
      "preco": 2.5,
      "id": 1,
      "nome": "Água"
    },
    {
      "preco": 5,
      "id": 2,
      "nome": "Coca"
    }
  ],
  "compras": [
    {
      "item_id": 1,
      "id": 1,
      "cliente_id": 1
    },
    {
      "item_id": 2,
      "id": 2,
      "cliente_id": 1
    }
  ]
}
```

Conclusão

A principal melhoria do projeto foi a substituição da persistência em arquivos JSON pelo uso de um banco de dados SQLite, o que permitiu uma maior robustez e consistência nos dados. Além disso, foram implementados novos endpoints para melhor gerenciamento de reservas e compras, além da geração de notas fiscais detalhadas. O sistema continua eficiente e modular, pronto para futuras expansões, como a integração com sistemas mais complexos de banco de dados ou uma interface gráfica para facilitar o uso por administradores do hotel.