

**Universidade Federal Rural de Pernambuco
Unidade Acadêmica de Serra Talhada
Curso: Sistemas de Informação**

Sistemas de Reserva de Hotel

Grupo:

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

Professor:

Michael Oliveira da Cruz

Serra Talhada - PE
Agosto, 2024

Sumário

Introdução3

Arquitetura3

Modelos de Dados3

Persistência de Dados3

Endpoints.....3

Imagens e Diagrama4

Conclusão6

Introdução

O projeto SISTEMA DE REGISTRO DE HOTEL consiste em uma aplicação backend desenvolvida com o framework FastAPI, cujo objetivo é de 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, utilizando arquivos JSON para a persistência dos dados relacionados aos quartos, clientes, reservas, itens e compras.

Arquitetura

A arquitetura do sistema foi estruturada de forma modular, utilizando o FastAPI como framework principal para a criação de APIs. A aplicação está organizada em modelos de dados, rotas para os diferentes endpoints e funções auxiliares responsáveis pelo carregamento e salvamento de dados.

Modelos de Dados

Os modelos de dados foram criados com o auxílio da biblioteca PYDANTIC, que oferece uma validação rigorosa e tipagem eficiente dos dados. 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.

Persistência de Dados

Os dados são persistidos em arquivo JSON, com funções específicas para os carregar e salvar enquanto executa a aplicação. Isso assegura que todas as modificações realizadas sejam mantidas entre as sessões de uso do sistema, proporcionando integridade e consistência nos dados geridos.

Endpoints

A API do sistema inclui vários endpoints que permitem a integração com as principais funcionalidades do sistema, como:

POST /quartos/: Adiciona novos quartos ao sistema.
 POST /clientes/: Registra novos clientes no sistema.
 POST /reservas/: Efetua a reserva de quartos.
 DELETE /reservas/{reserva_id}: Cancela uma reserva previamente realizada.
 PUT /quartos/{quarto_id}: Edita as informações de um quarto existente.
 POST /itens/: Adiciona novos itens ao inventário do hotel.
 POST /compras/: Registra compras realizadas por clientes.
 GET /dados_hotel/: Retorna um resumo de todos os dados armazenados, incluindo quartos, clientes, reservas, itens e compras.

Imagens e Diagrama

Figura 1 - Diagrama de Classe

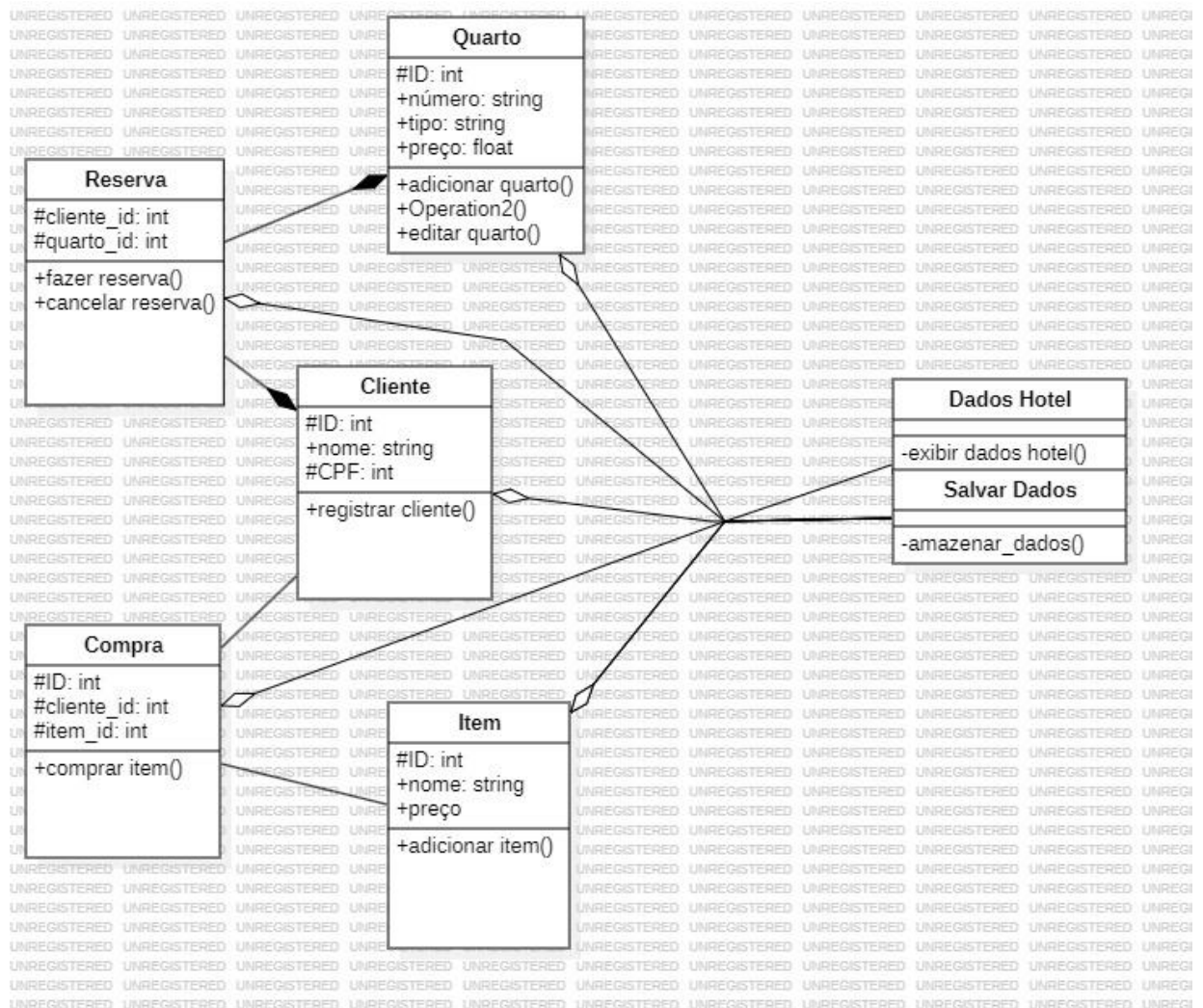


Figura 2 - Endpoints

default			^
POST	/quartos/	Adicionar Quarto	▼
POST	/clientes/	Registrar Cliente	▼
POST	/reservas/	Fazer Reserva	▼
DELETE	/reservas/{reserva_id}	Cancelar Reserva	▼
DELETE	/quartos/{quarto_id}	Remover Quarto	▼
PUT	/quartos/{quarto_id}	Editar Quarto	▼
POST	/itens/	Adicionar Item	▼
POST	/compras/	Comprar Item	▼
GET	/dados_hotel/	Exibir Dados Hotel	▼

Figura 3 - Adição de dados ao programa

POST

/quartos/

Adicionar Quarto

^

Parameters

Cancel

Reset

No parameters

Request body

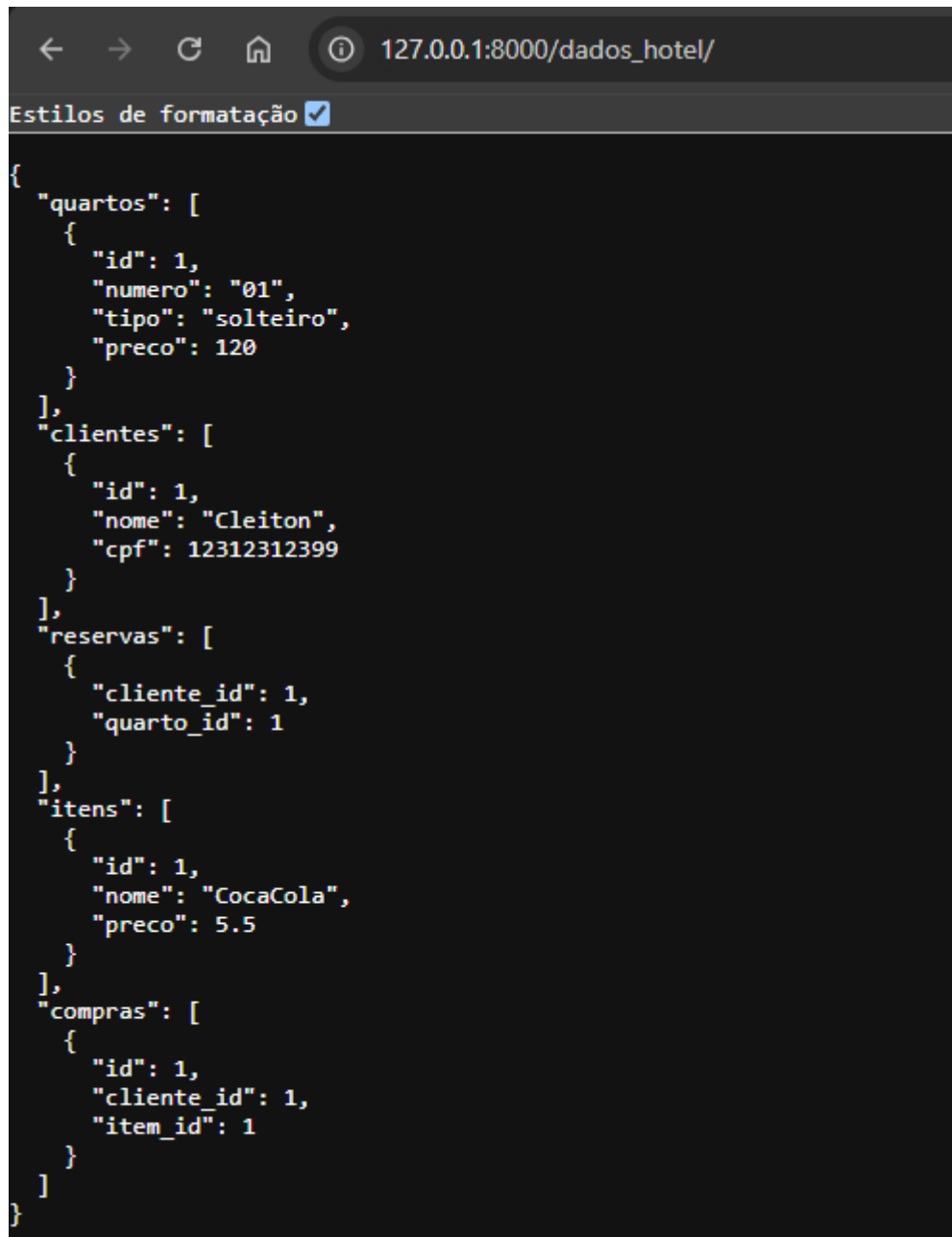
required

application/json

▼

```
{  "id": 0,  "numero": "01",  "tipo": "solteiro",  "preço": 120.00}
```

Figura 4 - Dados do Hotel com os dados



The image shows a web browser window with the address bar displaying '127.0.0.1:8000/dados_hotel/'. The page content shows a JSON response with the following structure:

```
{
  "quartos": [
    {
      "id": 1,
      "numero": "01",
      "tipo": "solteiro",
      "preco": 120
    }
  ],
  "clientes": [
    {
      "id": 1,
      "nome": "Cleiton",
      "cpf": 12312312399
    }
  ],
  "reservas": [
    {
      "cliente_id": 1,
      "quarto_id": 1
    }
  ],
  "itens": [
    {
      "id": 1,
      "nome": "CocaCola",
      "preco": 5.5
    }
  ],
  "compras": [
    {
      "id": 1,
      "cliente_id": 1,
      "item_id": 1
    }
  ]
}
```

Conclusão

O Sistema de Registro de Hotel atingiu com sucesso o objetivo de fornecer uma solução eficiente para o gerenciamento de um hotel, cobrindo operações críticas como o controle de reservas, gerenciamento de clientes, quartos, itens e compras. A utilização do FastAPI facilitou a construção de uma API robusta e ágil, enquanto a escolha por arquivos JSON para salvar e armazenar os dados ofereceu uma solução simples e eficaz para a persistência das informações. Como melhorias futuras, pode-se considerar a integração com um banco de dados para maior escalabilidade e a criação de uma interface gráfica para facilitar a utilização do sistema pelos usuários.