Texto

Descrição gerada automaticamente  
  
Mudança na estrutura do código para ficar mais organizado, agora tenho separado as partes de:  
Database – Banco de Dados  
Models – modelos definidos  
Routers – Rotas de cada tabela  
main.py – Arquivo principal que inicia tudo  
  
O que são as pastas \_\_pycache\_\_?  
Elas são o cache do Python, é onde ele guarda informações que ajudam seus programas a rodar mais rápido. Quando você executa um programa Python, ele traduz o código que você escreveu para uma linguagem que o computador entende melhor. Essa tradução pode levar algum tempo, então o Python guarda o resultado na pasta \_\_pycache\_\_ para usar na próxima vez que você rodar o programa, tornando tudo mais rápido. Você normalmente não precisa se preocupar com essa pasta; o Python cuida dela para você.  
  
**Começando pelo arquivo main.py:**  
Texto

Descrição gerada automaticamente  
Este código está criando um servidor de internet. Ele usa o FastAPI, que é uma ferramenta para construir servidores em Python. O servidor pode responder a pedidos de qualquer lugar (isso é chamado de “cross-origin”). Ele também define diferentes caminhos (ou “rotas”) que o servidor pode responder, que estão definidos em outro arquivo chamado chaves.

**Routers – chave.py:**Texto

Descrição gerada automaticamenteO código no arquivo routers define as rotas ou caminhos que o servidor pode responder. Cada rota é associada a uma função que é chamada quando a rota é acessada. As funções podem realizar várias ações, como buscar dados de um banco de dados ou adicionar novos dados. Neste caso, as rotas permitem listar todas as chaves, listar apenas as chaves disponíveis e adicionar uma nova chave. Cada rota retorna uma resposta que pode ser enviada de volta ao cliente que fez a solicitação.  
  
**Models – model\_chave.py:**Texto

Descrição gerada automaticamente

O arquivo models define a estrutura dos dados que o servidor pode receber ou enviar. Neste caso, ele define uma “Chave” como tendo um nome, uma situação e um status, todos sendo strings. Isso é usado para garantir que os dados enviados ao servidor estejam no formato correto e para informar ao servidor que tipo de dados ele deve esperar. Isso ajuda a evitar erros e torna o código mais fácil de entender e manter.  
  
**Database – connection.py:**  
Texto

Descrição gerada automaticamente

O arquivo connection contém uma função que estabelece uma conexão com um banco de dados PostgreSQL. Ele usa a biblioteca psycopg2 para criar essa conexão. A função retorna a conexão que pode ser usada para executar consultas SQL no banco de dados. Os detalhes do banco de dados, como host, nome do banco de dados, nome do usuário e senha, são fornecidos ao criar a conexão.