**Criando Tabela no Banco de Dados**

O banco de dados que vem com o Django é o sqlite3. Porém, o sqlite3 é um banco de dados que não roda em servidor, ele é apenas um arquivo na sua máquina. Dessa forma, você pode desenvolver o projeto inteiro com o **sqlite3** e depois quando precisar subir o projeto para produção você pode fazer a configuração da variável **DATABASES** no **settings.py** com o banco de dados escolhido (MySQL, Postgres, etc).

Comandos **makemigrations** e **migrate**

**python manage.py makemigrations →**  detecta se houve alguma mudança no banco de dados

**python manage.py migrate →**  aplica as migrações do banco de dados

Por padrão, o Django já vem com uma série de tabelas no banco de dados. Para visualizá-las basta dar o comando **migrate**. Se você quiser criar suas próprias tabelas personalizadas para o banco de dados você deverá fazer a modelagem das mesmas no **models.py** da aplicação e em seguida dar os comandos **makemigrations** para criar os arquivos de migração com as alterações feitas e o **migrate** para aplicar as migrações.

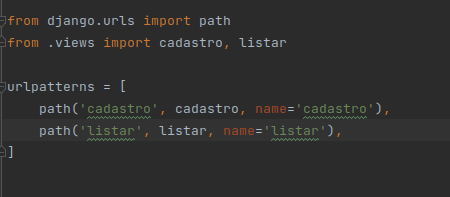
Inserindo valores na tabela do banco de dados

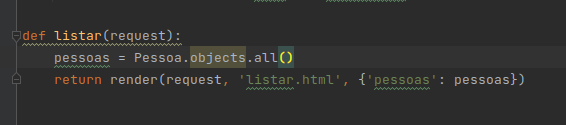
Para que as informações do formulário sejam salvas no banco de dados precisa importar a classe referente a tabela que foi modelada no **models.py**, instanciar a classe em **views.py** e salvar os dados com **.save():**

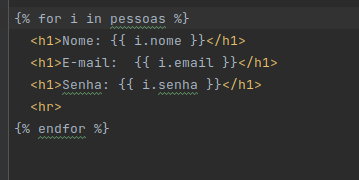


**Trazendo informações do Banco de dados pro Template**

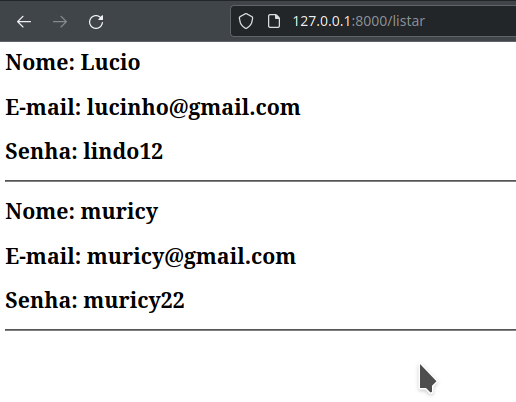
Para trazer as informações cadastradas no banco de dados para uma página **HTML** você precisa definir a **URL** que irá ser acessada para a página, criar a função em **views.py** e criar o respectivo **template**:



 Em **listar.html**:



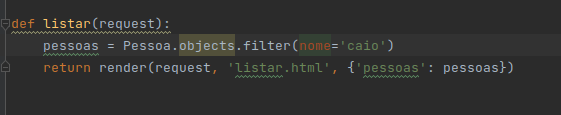
Dessa forma, irá aparecer na página html as informações cadastradas no banco de dados:

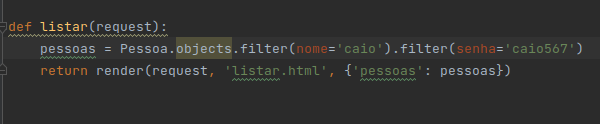


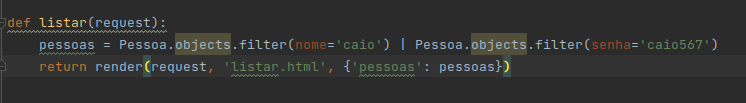
**Filtrando objetos do banco de dados**

No exemplo anterior foi mostrado como exibir as informações de todos os usuários cadastrados no banco de dados. Nós também podemos fazer o filtro e exibir do banco de dados apenas as informações de acordo com condições específicas. Para isso, no lugar de **.all()** que exibe todos os objetos, utiliza-se o **.filter()** com a condição desejada dentro dos parênteses.

No exemplo abaixo só serão exibidas na página HTML as informações do usuário que tiver **caio** como **nome:**

Já no exemplo a seguir serão exibidas na página HTML as informações do usuário que tiver ao mesmo tempo o nome **caio** e a senha **caio567:**

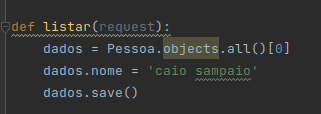
No exemplo seguinte irá ser retornado para a página **HTML** as informação do usuário de nome caio **OU** o usuário com a senha caio567:



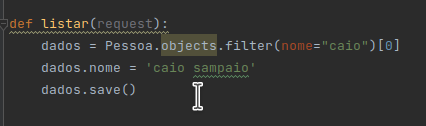
**Fazer Update do Banco de Dados**

Para fazer a modificação de um dado no banco de dados existem diferentes formas

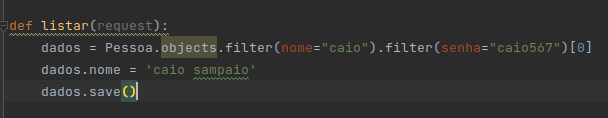
1ª forma:



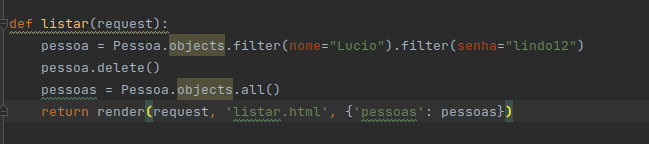
2ª forma:



3ª forma:

**Deletando um dado do banco de dados**

Para deletar um dado do banco de dados faça:

Nesse exemplo o dado especificado no filtro é deletado sendo mostrados, após isso, o restante dos dados presentes no banco de dados