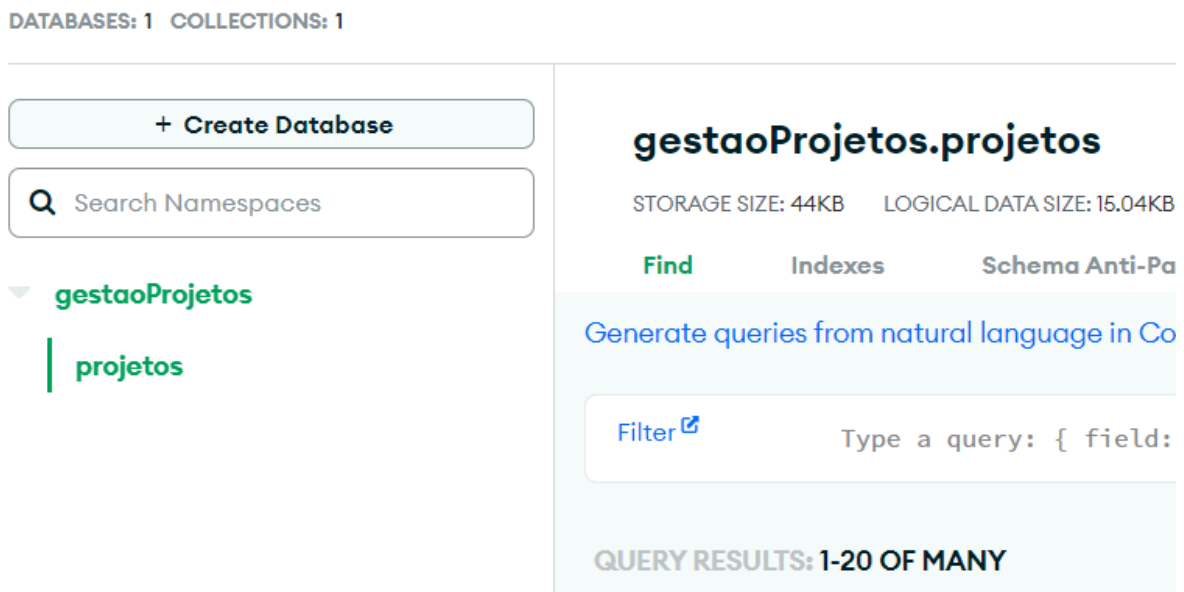


APS 1 - Igor André Sona Martins e Matheus Fajardo Labres Bueno

1 - Contexto



66 documentos na coleção! "projeto" com sucesso!

2 - Operações

a) Adicionar um novo documento na coleção projeto.

```
novo_documento = {
    "nome": "Desenvolvimento de Aplicativo Móvel",
    "metodologia": "Ágil",
    "tecnologias": ["Kotlin", "Swift", "React Native"],
    "orcamento": 50000,
    "data_conclusao": "2024-12-31",
    "lider_projeto": "Sandro de Araujo"
}
resultado_insercao = collection.insert_one(novo_documento)
print(f"Novo documento inserido com sucesso! ID: {resultado_insercao.inserted_id}")
```

Novo documento inserido com sucesso! ID: 66cd0547d66a8b85e17dc957

```
_id: ObjectId('66cd0547d66a8b85e17dc957')
nome: "Desenvolvimento de Aplicativo Móvel"
metodologia: "Ágil"
▼ tecnologias: Array (3)
  0: "Kotlin"
  1: "Swift"
  2: "React Native"
orcamento: 50000
data_conclusao: "2024-12-31"
lider_projeto: "Sandro de Araujo"
```

b) Consultar todos os documentos na coleção projetos, e que tenham como metodologia o método Ágil.

Documentos na coleção projetos com a metodologia Ágil

```
{'nome': 'Desenvolvimento de Sistema de E-commerce'}
{'nome': 'Plataforma de Ensino a Distância'}
{'nome': 'Desenvolvimento de API RESTful'}
{'nome': 'Automação Residencial'}
{'nome': 'Plataforma de Jogos Online'}
{'nome': 'Portal de Notícias'}
{'nome': 'Plataforma de Streaming de Vídeo'}
{'nome': 'Desenvolvimento de Chatbot'}
{'nome': 'Desenvolvimento de Sistema de Reservas'}
{'nome': 'Plataforma de Social Media'}
{'nome': 'Sistema de Logística Integrada'}
{'nome': 'Sistema de Gestão de Frota'}
{'nome': 'Plataforma de Consultoria Financeira'}
{'nome': 'Plataforma de Telemedicina'}
{'nome': 'Plataforma de Gestão de Eventos'}
{'nome': 'Sistema de Monitoramento Ambiental'}
{'nome': 'Sistema de Gestão de Inventário'}
{'nome': 'Desenvolvimento de Plataforma de Marketing Digital'}
{'nome': 'Plataforma de Treinamento Online'}
{'nome': 'Sistema de Gestão de Transporte Público'}
{'nome': 'Aplicativo de Mobilidade Urbana'}
{'nome': 'Sistema de Análise de Dados de IoT'}
{'nome': 'Plataforma de Suporte a Clientes'}
{'nome': 'Automação de Processos Logísticos'}
{'nome': 'Plataforma de Treinamento de Inteligência Artificial'}
{'nome': 'Desenvolvimento de API para E-commerce'}
{'nome': 'Desenvolvimento de Aplicativo Móvel'}
```

c) Atualizar o valor do orçamento para o projeto adicionado, alterando o valor para R\$ 60.000,00. Posteriormente, atualizar a data de conclusão (30/11/2024) e o líder (Judite Glavatzki).

Atualizando orçamento

```
filtro = {"nome": "Desenvolvimento de aplicativo móvel"}
novo_valor = {"$set": {"orcamento": 60000}}
resultado_atualizacao = collection.update_one(filtro, novo_valor)
if resultado_atualizacao.modified_count > 0:
    print("O orçamento foi atualizado")
else:
    print("Nenhum documento foi atualizado.")
```

```
PS C:\Users\Aluno\Desktop\aps> & C:/Python311/python.exe c:/Users/Aluno/Desktop/aps/crud.py
O orçamento foi atualizado
PS C:\Users\Aluno\Desktop\aps> █
```

Atualizando data de conclusão

```
filtro = {"nome": "Desenvolvimento de aplicativo móvel"}
novo_valor = {"$set": {"data_conclusao": "2024-11-30"}}
resultado_atualizacao = collection.update_one(filtro, novo_valor)
if resultado_atualizacao.modified_count > 0:
    print("A data foi atualizada")
else:
    print("Nenhum documento foi atualizado.")
```

```
PS C:\Users\Aluno\Desktop\aps> & C:/Python311/python.exe c:/Users/Aluno/Desktop/aps/crud.py
A data foi atualizada
PS C:\Users\Aluno\Desktop\aps> █
```

Atualizando líder do projeto

```
filtro = {"nome": "Desenvolvimento de Aplicativo Móvel"}
novo_valor = {"$set": {"lider_projeto": "Judite Glavatzki"}}
resultado_atualizacao = collection.update_one(filtro, novo_valor)
if resultado_atualizacao.modified_count > 0:
    print("Documento atualizado com sucesso!")
else:
    print("Nenhum dado foi atualizado!")
```

Documento atualizado com sucesso!

```
_id: ObjectId('66cd0547d66a8b85e17dc957')
nome: "Desenvolvimento de Aplicativo Móvel"
metodologia: "Ágil"
▼ tecnologias: Array (3)
  0: "Kotlin"
  1: "Swift"
  2: "React Native"
orcamento: 60000
data_conclusao: "2024-11-30"
lider_projeto: "Judite Glavatzki"
```

d) Remover todos os projetos com orçamento menor que R\$ 50.000.

```
filtro = {"orcamento": {"$lt": 50000}}
resultado_delete = collection.delete_many(filtro)
if resultado_delete.deleted_count > 0:
    print("Documentos deletados com sucesso!")
else:
    print("Nenhum documento foi deletado!")
```

Nenhum documento foi deletado!

3 - Consultas

- a) Projetos com orçamento acima de R\$ 100.000

```
''' Query 1 '''
filtro = {"orcamento": {"$gt": 100000}}
find = collection.find(filtro, {"nome":1, "_id":0, "orcamento":1})
data = list(find)
df = pd.DataFrame(data)
novo_cabecalho = ["Nome do projeto", "Orçamento"]
print(tabulate(df, headers=novo_cabecalho, tablefmt="fancy_grid"))
```

| | Nome do projeto | Orçamento |
|----|--|-----------|
| 0 | Desenvolvimento de Sistema de E-commerce | 150000 |
| 1 | Plataforma de Ensino a Distância | 120000 |
| 2 | Sistema de Gestão Hospitalar | 200000 |
| 3 | Desenvolvimento de API RESTful | 110000 |
| 4 | Plataforma de Jogos Online | 130000 |
| 5 | Sistema de Votação Online | 120000 |
| 6 | Plataforma de Streaming de Vídeo | 200000 |
| 7 | Sistema de Gestão de Projetos | 105000 |
| 8 | Sistema de Pagamentos Online | 170000 |
| 9 | Sistema de Gestão de Recursos Humanos | 125000 |
| 10 | Plataforma de Social Media | 160000 |
| 11 | Sistema de Logística Integrada | 140000 |
| 12 | Automação de Processos Industriais | 180000 |
| 13 | Plataforma de Leilões Online | 105000 |
| 14 | Sistema de Rastreamento de Entregas | 110000 |
| 15 | Plataforma de Consultoria Financeira | 125000 |
| 16 | Plataforma de Telemedicina | 155000 |
| 17 | Automação de Processos Jurídicos | 115000 |
| 18 | Plataforma de E-learning Corporativo | 130000 |
| 19 | Sistema de Gestão de Bibliotecas | 125000 |
| 20 | Plataforma de Treinamento Online | 155000 |
| 21 | Desenvolvimento de Plataforma de Comércio B2B | 130000 |
| 22 | Sistema de Gestão de Transporte Público | 140000 |
| 23 | Sistema de Gestão de Energia Sustentável | 160000 |
| 24 | Aplicativo de Mobilidade Urbana | 105000 |
| 25 | Sistema de Análise de Dados de IoT | 105000 |
| 26 | Plataforma de Suporte a Clientes | 115000 |
| 27 | Automação de Processos Logísticos | 150000 |
| 28 | Sistema de Análise de Dados para Marketing | 120000 |
| 29 | Plataforma de Treinamento de Inteligência Artificial | 200000 |
| 30 | Sistema de Monitoramento de Energia Solar | 110000 |
| 31 | Desenvolvimento de API para E-commerce | 130000 |
| 32 | Automação de Processos Manufatureiros | 180000 |
| 33 | Sistema de Controle de Qualidade de Produtos | 125000 |

b) Consultas de projetos que usam tecnologias baseadas em Firebase

```
filtro = {"tecnologias": "Firebase"}
find = collection.find(filtro, {"nome":1, "_id":0})
data = list(find)
df = pd.DataFrame(data)
print(tabulate(df, headers="keys", tablefmt="grid"))
```

| | nome |
|---|--|
| 0 | Aplicativo Mobile para Fintech |
| 1 | Automação Residencial |
| 2 | Plataforma de Jogos Online |
| 3 | Aplicativo de Entrega de Comida |
| 4 | Aplicativo de Reservas de Viagens |
| 5 | Sistema de Monitoramento Ambiental |
| 6 | Aplicativo de Monitoramento de Energia |
| 7 | Aplicativo de Mobilidade Urbana |
| 8 | Aplicativo de Gestão de Dietas |

c) Projetos com orçamento entre R\$80.000 e R\$100.000.

```
filtro = {"orcamento": {"$gt": 80000, "$lt": 100000}}
find = collection.find(filtro, {"nome":1, "_id":0, "orcamento":1})
data = list(find)
df = pd.DataFrame(data)
novo_cabecalho = ["Nome do projeto", "Orçamento"]
print(tabulate(df, headers=novo_cabecalho, tablefmt="fancy_grid"))
```

| | Nome do projeto | Orçamento |
|----|--|-----------|
| 0 | Integração de Sistemas Legados | 95000 |
| 1 | Sistema de Atendimento ao Cliente | 85000 |
| 2 | Aplicativo de Monitoramento de Saúde | 95000 |
| 3 | Plataforma de Cursos Online | 90000 |
| 4 | Aplicativo de Gerenciamento de Finanças Pessoais | 85000 |
| 5 | Sistema de Gestão de Frota | 95000 |
| 6 | Aplicativo de Reservas de Viagens | 98000 |
| 7 | Sistema de Auditoria Contábil | 90000 |
| 8 | Desenvolvimento de Aplicativo para Fitness | 85000 |
| 9 | Sistema de Gestão de Inventário | 85000 |
| 10 | Desenvolvimento de Plataforma de Marketing Digital | 98000 |
| 11 | Plataforma de Gestão de Cursos Online | 95000 |
| 12 | Desenvolvimento de Aplicativo Financeiro | 85000 |
| 13 | Sistema de Gestão de Estoque | 90000 |
| 14 | Desenvolvimento de Aplicativo para Educação Infantil | 95000 |
| 15 | Aplicativo de Gestão de Dietas | 85000 |
| 16 | Plataforma de Gerenciamento de Tarefas | 95000 |

d) Projetos que têm a metodologia Scrum e orçamento acima de R\$100.000.

```
filtro = {"orcamento": {"$gt": 100000}, "metodologia": "Scrum"}
find = collection.find(filtro, {"nome":1, "_id":0, "metodologia":1})
data = list(find)
df = pd.DataFrame(data)
novo_cabecalho = ["Nome do projeto", "Metodologia"]
print(tabulate(df, headers=novo_cabecalho, tablefmt="fancy_grid"))
```

| | Nome do projeto | Metodologia |
|---|--|-------------|
| 0 | Sistema de Gestão Hospitalar | Scrum |
| 1 | Sistema de Gestão de Recursos Humanos | Scrum |
| 2 | Plataforma de Leilões Online | Scrum |
| 3 | Plataforma de E-learning Corporativo | Scrum |
| 4 | Sistema de Controle de Qualidade de Produtos | Scrum |

e) Projetos concluídos antes de 01/01/2024.

```
filtro = {"data_conclusao": {"$lt": "2024-01-01"}}
find = collection.find(filtro,{"nome": 1, "_id":0, "data_conclusao": 1})
data = list(find)
df = pd.DataFrame(data)
novo_cabecalho = ["Nome do projeto", "Data da conclusão"]
print(tabulate(df, headers=novo_cabecalho, tablefmt="fancy_grid"))
```


| | Nome do projeto | Data da conclusão |
|----|--|-------------------|
| 0 | Desenvolvimento de Sistema de E-commerce | 2023-05-20 |
| 1 | Automação de Processos Financeiros | 2022-12-15 |
| 2 | Plataforma de Ensino a Distância | 2023-08-10 |
| 3 | Aplicativo Mobile para Fintech | 2023-06-25 |
| 4 | Integração de Sistemas Legados | 2023-11-05 |
| 5 | Desenvolvimento de API RESTful | 2023-04-18 |
| 6 | Sistema de Controle de Estoque | 2022-10-12 |
| 7 | Plataforma de Jogos Online | 2023-07-10 |
| 8 | Sistema de Atendimento ao Cliente | 2023-09-20 |
| 9 | Aplicativo de Monitoramento de Saúde | 2023-03-05 |
| 10 | Portal de Notícias | 2022-11-30 |
| 11 | Plataforma de Streaming de Vídeo | 2023-12-22 |
| 12 | Sistema de Gestão de Projetos | 2023-01-25 |
| 13 | Desenvolvimento de Chatbot | 2023-08-14 |
| 14 | Sistema de Pagamentos Online | 2023-10-02 |
| 15 | Plataforma de Cursos Online | 2023-11-29 |
| 16 | Desenvolvimento de Sistema de Reservas | 2022-08-19 |
| 17 | Sistema de Gestão de Recursos Humanos | 2023-06-21 |
| 18 | Desenvolvimento de Sistema de Tickets | 2023-12-01 |
| 19 | Sistema de Logística Integrada | 2023-07-15 |
| 20 | Automação de Processos Industriais | 2023-09-01 |
| 21 | Plataforma de Leilões Online | 2023-11-10 |
| 22 | Sistema de Rastreamento de Entregas | 2023-10-18 |
| 23 | Sistema de Gestão Escolar | 2023-05-25 |
| 24 | Aplicativo de Reservas de Viagens | 2023-08-23 |
| 25 | Sistema de Auditoria Contábil | 2023-11-29 |

| | | |
|----|--|------------|
| 26 | Plataforma de Gestão de Eventos | 2023-12-20 |
| 27 | Automação de Processos Jurídicos | 2023-09-19 |
| 28 | Sistema de Gestão de Inventário | 2023-10-11 |
| 29 | Desenvolvimento de Plataforma de Marketing Digital | 2023-08-10 |
| 30 | Sistema de Gestão de Bibliotecas | 2023-06-20 |
| 31 | Sistema de Controle de Qualidade | 2023-09-30 |
| 32 | Desenvolvimento de Sistema de Atendimento ao Cliente | 2023-10-12 |
| 33 | Aplicativo de Mobilidade Urbana | 2023-11-07 |
| 34 | Plataforma de Gestão de Cursos Online | 2023-12-21 |
| 35 | Sistema de Gestão de Estoque | 2023-11-30 |

f) Projetos com metodologia Waterfall que usam SQL Server.

| | Nome do projeto | Metodologia | Tecnologia |
|---|---|-------------|------------------------------|
| 0 | Sistema de Gestão Escolar | Waterfall | ['C#', '.NET', 'SQL Server'] |
| 1 | Automação de Processos Jurídicos | Waterfall | ['C#', '.NET', 'SQL Server'] |
| 2 | Sistema de Gestão de Bibliotecas | Waterfall | ['C#', '.NET', 'SQL Server'] |
| 3 | Sistema de Monitoramento de Energia Solar | Waterfall | ['C#', '.NET', 'SQL Server'] |

```
filtro = {"metodologia": "Waterfall", "tecnologias": "SQL Server"}
find = collection.find(filtro, {"nome":1, "_id":0, "tecnologias": 1, "metodologia": 1})
data = list(find)
df = pd.DataFrame(data)
novo_cabecalho = ["Nome do projeto", "Metodologia", "Tecnologia"]
print(tabulate(df, headers=novo_cabecalho, tablefmt="fancy_grid"))
```

g) Projetos com o nome que contém a palavra Sistema. Utilize o operador regex.

```
pattern = re.compile(r'Sistema', re.IGNORECASE)
result = collection.find({"nome": pattern}, {"nome":1, "_id":0})
data = list(result)
df = pd.DataFrame(data)
novo_cabecalho = ["Nome do projeto"]
print(tabulate(df, headers=novo_cabecalho, tablefmt="fancy_grid"))
```

| | Nome do projeto |
|----|--|
| 0 | Desenvolvimento de Sistema de E-commerce |
| 1 | Sistema de Gestão Hospitalar |
| 2 | Integração de Sistemas Legados |
| 3 | Sistema de Controle de Estoque |
| 4 | Sistema de Atendimento ao Cliente |
| 5 | Sistema de Votação Online |
| 6 | Sistema de Gestão de Projetos |
| 7 | Sistema de Pagamentos Online |
| 8 | Desenvolvimento de Sistema de Reservas |
| 9 | Sistema de Gestão de Recursos Humanos |
| 10 | Desenvolvimento de Sistema de Tickets |
| 11 | Sistema de Logística Integrada |
| 12 | Sistema de Gestão de Frota |
| 13 | Sistema de Rastreamento de Entregas |
| 14 | Sistema de Gestão Escolar |
| 15 | Sistema de Auditoria Contábil |
| 16 | Sistema de Monitoramento Ambiental |
| 17 | Sistema de Gestão de Inventário |
| 18 | Sistema de Gestão de Bibliotecas |
| 19 | Sistema de Controle de Qualidade |
| 20 | Sistema de Gestão de Transporte Público |
| 21 | Desenvolvimento de Sistema de Atendimento ao Cliente |
| 22 | Sistema de Gestão de Energia Sustentável |
| 23 | Sistema de Análise de Dados de IoT |

| | |
|----|--|
| 24 | Sistema de Gestão de Estoque |
| 25 | Sistema de Análise de Dados para Marketing |
| 26 | Sistema de Monitoramento de Energia Solar |
| 27 | Sistema de Controle de Qualidade de Produtos |

h) Projetos que utilizam as tecnologias Python e Flask. Utilize o operador all.

| | Nome do projeto | Tecnologias |
|---|--|-----------------------------------|
| 0 | Integração de Sistemas Legados | ['Python', 'Flask', 'PostgreSQL'] |
| 1 | Plataforma de Cursos Online | ['Python', 'Flask', 'PostgreSQL'] |
| 2 | Plataforma de Consultoria Financeira | ['Python', 'Flask', 'MySQL'] |
| 3 | Desenvolvimento de Plataforma de Marketing Digital | ['Python', 'Flask', 'MySQL'] |
| 4 | Sistema de Análise de Dados de IoT | ['Python', 'Flask', 'MongoDB'] |
| 5 | Desenvolvimento de API para E-commerce | ['Python', 'Flask', 'MySQL'] |

```
filtro = {"tecnologias": {"$all" : ["Python", "Flask"]}}
resultados = collection.find(filtro, {"nome":1, "tecnologias":1, "_id":0})
data = list(resultados)
df = pd.DataFrame(data)
novo_cabecalho = ["Nome do projeto", "Tecnologias"]
print(tabulate(df, headers=novo_cabecalho, tablefmt="fancy_grid"))
```

i) Projetos liderados por Adrian Netto ou Allan Pacheco. Utilize o operador in.

| | Nome do projeto | Lider do projeto |
|---|--|------------------|
| 0 | Desenvolvimento de Sistema de Atendimento ao Cliente | Allan Pacheco |
| 1 | Aplicativo de Mobilidade Urbana | Adrian Netto |

```
filtro = {"lider_projeto": {"$in" : ["Adrian Netto", "Allan Pacheco"]}}
resultados = collection.find(filtro, {"nome":1, "lider_projeto":1, "_id":0})
data = list(resultados)
df = pd.DataFrame(data)
novo_cabecalho = ["Nome do projeto", "Lider do projeto"]
print(tabulate(df, headers=novo_cabecalho, tablefmt="fancy_grid"))
```

j) Projetos que utilizam "Node.js" e foram finalizados antes de 2024.

| | Nome do projeto | Tecnologias | Data de conclusão |
|---|--|--------------------------------------|-------------------|
| 0 | Plataforma de Ensino a Distância | ['Node.js', 'React', 'MongoDB'] | 2023-08-10 |
| 1 | Portal de Notícias | ['Node.js', 'React', 'MongoDB'] | 2022-11-30 |
| 2 | Desenvolvimento de Chatbot | ['Node.js', 'Express', 'MongoDB'] | 2023-08-14 |
| 3 | Sistema de Rastreamento de Entregas | ['Node.js', 'Express', 'MongoDB'] | 2023-10-18 |
| 4 | Desenvolvimento de Sistema de Atendimento ao Cliente | ['Node.js', 'Express', 'MongoDB'] | 2023-10-12 |
| 5 | Sistema de Gestão de Estoque | ['JavaScript', 'Node.js', 'MongoDB'] | 2023-11-30 |

```
filtro = {"tecnologias": "Node.js", "data_conclusao": {"$lt": "2024-01-01"}}
resultado = collection.find(filtro, {"nome":1, "data_conclusao":1, "tecnologias":1, "_id":0})
data = list(resultado)
df = pd.DataFrame(data)
novo_cabecalho = ["Nome do projeto", "Tecnologias", "Data de conclusão"]
print(tabulate(df, headers=novo_cabecalho, tablefmt="fancy_grid"))
```

k) Média de orçamento dos projetos que desenvolvidos com a metodologia Ágil.

```
''' Query 11 '''
pipeline = [
    {"$match": {"metodologia": "Ágil"}},
    {"$group": {"_id":None, "media_metodologia": {"$avg": "$orcamento"}}}
]
resultado = list(collection.aggregate(pipeline))
if resultado:
    media_metodologia = resultado[0]["media_metodologia"]
    print(f"A média dos projetos com a metodologia ágil é {media_metodologia:.0f}")
```

A média de orçamento dos projetos com metodologia Ágil é de R\$116.222,00

l) Todos os projetos desenvolvidos com a metodologia Waterfall

```
filtro = {"metodologia": "Waterfall"}
resultado = collection.find(filtro, {"_id":0, "nome":1, "metodologia":1})
data = list(resultado)
df = pd.DataFrame(data)
novo_cabecalho = ["Nome do projeto", "Metodologia"]
print(tabulate(df, headers=novo_cabecalho, tablefmt="fancy_grid"))
```

| | Nome do projeto | Metodologia |
|----|---|-------------|
| 0 | Automação de Processos Financeiros | Waterfall |
| 1 | Integração de Sistemas Legados | Waterfall |
| 2 | Sistema de Votação Online | Waterfall |
| 3 | Sistema de Pagamentos Online | Waterfall |
| 4 | Desenvolvimento de Sistema de Tickets | Waterfall |
| 5 | Automação de Processos Industriais | Waterfall |
| 6 | Sistema de Gestão Escolar | Waterfall |
| 7 | Sistema de Auditoria Contábil | Waterfall |
| 8 | Automação de Processos Jurídicos | Waterfall |
| 9 | Sistema de Gestão de Bibliotecas | Waterfall |
| 10 | Desenvolvimento de Plataforma de Comércio B2B | Waterfall |
| 11 | Sistema de Gestão de Energia Sustentável | Waterfall |
| 12 | Desenvolvimento de Aplicativo Financeiro | Waterfall |
| 13 | Sistema de Monitoramento de Energia Solar | Waterfall |
| 14 | Automação de Processos Manufatureiros | Waterfall |

m) Qual o projeto com o maior orçamento?

O projeto com maior orçamento dentro da base de dados é “Sistema de Gestão Hospitalar” com um orçamento de R\$200.000,00.

```
''' Query 13 '''
filtro = collection.find({}, {"nome":1, "orcamento":1, "_id":0}).sort("orcamento", -1).limit(1)
for maiorOrcamento in filtro:
    print(maiorOrcamento)
```

- n) Qual a metodologia com o maior número de projeto? Combinar os operadores unwind, group, sort e limit

```
pipeline = [
    {"$group": {"_id": "$metodologia", "count": {"$sum": 1}}},
    {"$sort": {"count": -1}},
    {"$limit": 1}
]
result = list(collection.aggregate(pipeline))
if result:
    top_metodologia = result[0]
    print(f'Metodologia com maior número de projetos: {top_metodologia["_id"]} com {top_metodologia["count"]} projetos')
else:
    print('Nenhum dado encontrado.')
```

Metodologia com maior número de projetos: Ágil com 27 projetos

- o) Todos os projetos que utilizam Ruby e Rails como tecnologias.

```
''' Query 15'''
filtro = {"tecnologias": {"$all": ["Ruby", "Rails"]}}
resultado = collection.find(filtro, {"nome":1, "tecnologias":1, "_id":0})
data = list(resultado)
df = pd.DataFrame(data)
novo_cabecalho = ["Nome do projeto", "Tecnologias"]
print(tabulate(df, headers=novo_cabecalho, tablefmt="fancy_grid"))
```

| Nome do projeto | Tecnologias |
|--|-----------------------------|
| Sistema de Controle de Estoque | ['Ruby', 'Rails', 'SQLite'] |
| Sistema de Gestão de Projetos | ['Ruby', 'Rails', 'SQLite'] |
| Plataforma de Leilões Online | ['Ruby', 'Rails', 'MySQL'] |
| Plataforma de Gestão de Cursos Online | ['Ruby', 'Rails', 'MySQL'] |
| Plataforma de Gerenciamento de Tarefas | ['Ruby', 'Rails', 'SQLite'] |

- p) Quais são os projetos desenvolvidos com Kanban e que têm um orçamento menor que R\$ 80.000?

```
''' Query 16'''
filtro = {"metodologia": "Kanban", "orcamento": {"$lt": 80000}}
resultado = collection.find(filtro, {"nome":1, "metodologia":1, "orcamento":1, "_id":0})
data = list(resultado)
df = pd.DataFrame(data)
novo_cabecalho = ["Nome do projeto", "Metodologia", "Orçamento"]
print(tabulate(df, headers=novo_cabecalho, tablefmt="fancy_grid"))
```

| Nome do projeto | Metodologia | Orçamento |
|--------------------------------|-------------|-----------|
| Aplicativo Mobile para Fintech | Kanban | 70000 |