

DESAFIO PUBFUTURE

DOCUMENTAÇÃO API CONTROLE FINANCEIRO

Autor

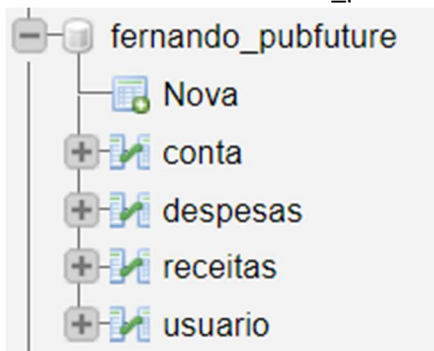
Fernando Artur Luithardt

SUMÁRIO

PROCEDIMENTOS UTILIZAR A API.....	2
USUÁRIO.....	5
CONTA.....	7
DESPESAS.....	10
RECEITAS.....	16

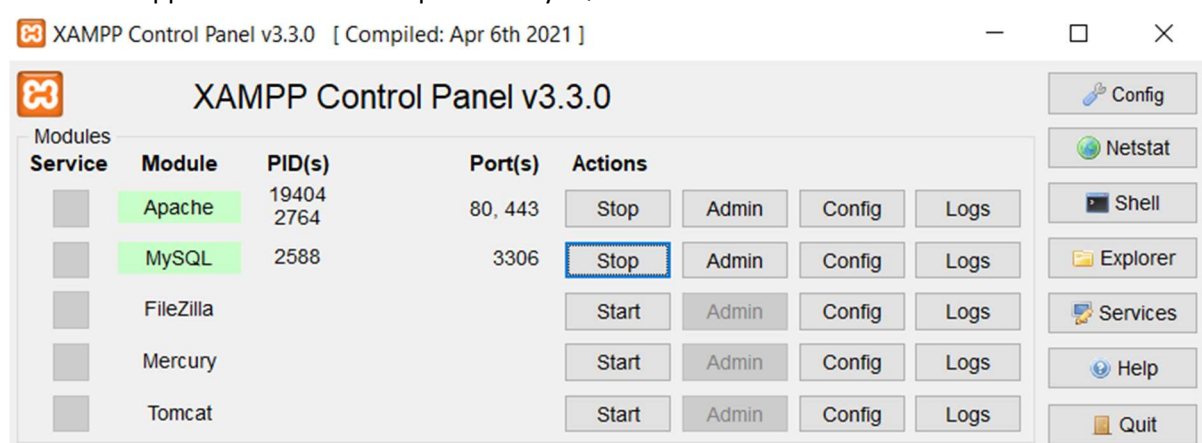
PROCEDIMENTOS PARA UTILIZAR A API

1. Os arquivos ficam em "C:\xampp\htdocs\DesafioPubFuture\arquivos do github"
2. Foi utilizado o software XAMP com Apache e MySql, no PhpMyAdmin é necessário criar uma base chamada "fernando_pubfuture" a importar o arquivo "fernando_pubfuture.sql"

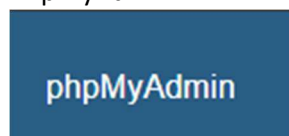


Como criar base de dados:

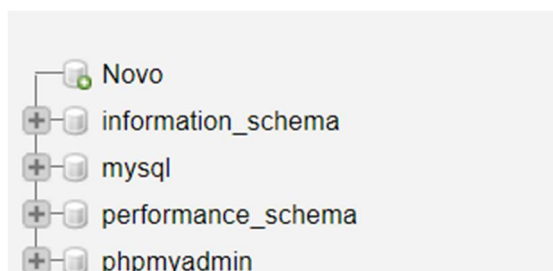
Inicie o xampp e inicie o módulo Apache e MySQL



Digite "localhost" no seu navegador e ao abrir a página no canto superior direito abra PhpMyAdmin:






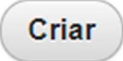
Ao Abrir a página phpMyAdmin no canto esquerdo clique em "novo"



Ao abrir a nova tela preencha o campo "nome da base de dados" com "fernando_pubfuture", em seguida aperte em criar

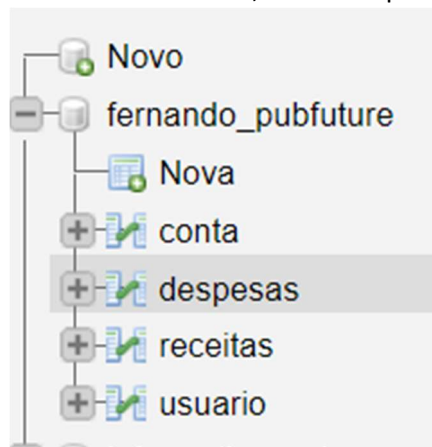
Base de Dados

 Criar base de dados 

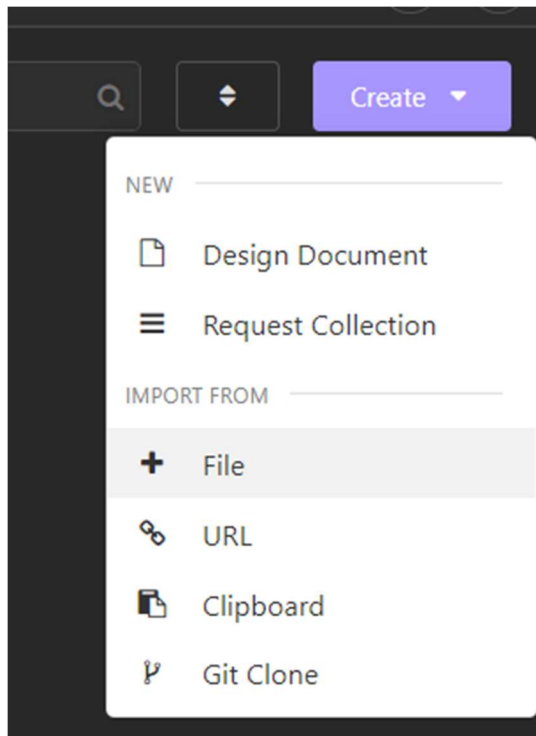
 

Base de Dados  **Agrupamento (Collation)** **Ações**

Ao fazer isso será aberta uma nova base, clique na aba “importar”, e na opção escolher arquivo escolha o arquivo “Fernando_pubfuture.sql”, após isto rolar até o final da página e clicar em “executar”, está tudo pronto e o banco ficará desta forma a seguir:



3. Para teste da API foi utilizado o software “INSOMNIA”, sendo que enviando dados via POST utilizando estrutura ‘MultipartForm’, para facilitar o seu já exportei o arquivo “Insomnia.json”, que pode ser importando para o insomnia contado todos endereços da api, para importar abre o insomnia, no canto direito superior clique em create > file (escolha o arquivo insomnia.json)



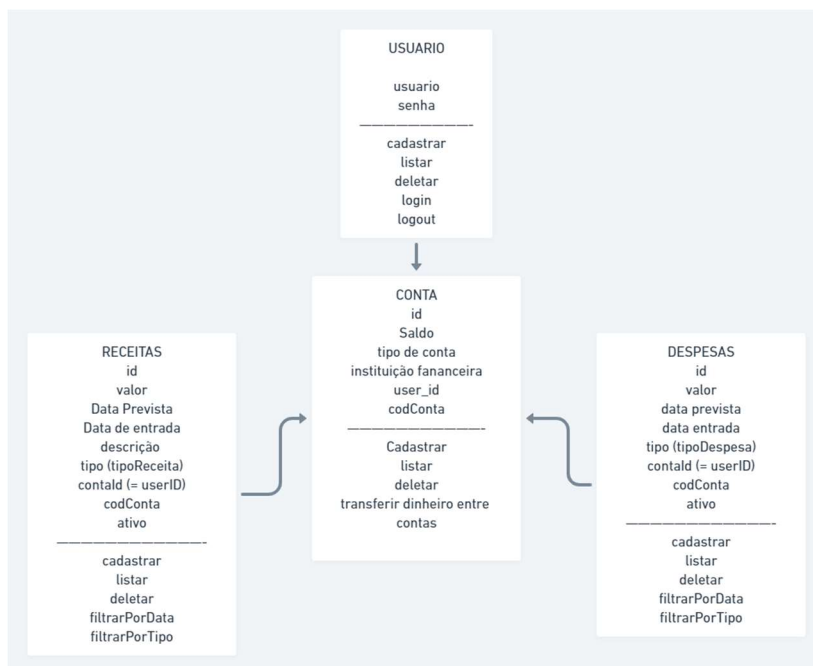
Após importar o arquivo “insomnia.json”, abra o projeto e fique livre para utilizar a api, lembrando que é necessário inicializar o xampp e estar com a base de dados inserida nele. No arquivo “fernando_pubfuture.sql” já deixei criado dois usuários com contas, despesas e receitas associados.

Usuario: PUBFUTURE

Senha: PUBFUTURE

codConta: 12345 (onde tem mais receitas/despesas cadastrados)

codConta: 123456 (cadastrei menos receitas/ despesas itens nesta conta)



Base de dados resumida

USUARIOS:

CRIANDO USUÁRIO:

API: <http://localhost/DesafioPubFuture/usuario/cadastrar/>

Método: POST

Necessário enviar: usuario, senha

RESPOSTAS:

{"mensagem":"OK"} Cod(200) (usuario criado com sucesso)

{"mensagem":"Usuario ja existente"} Cod(400) (nome de usuario já existente no banco)

LOGIN USUÁRIO:

API: <http://localhost/DesafioPubFuture/usuario/login/>

Método: POST

Necessário enviar: usuario, senha

RESPOSTAS:

{"mensagem":"OK"} Cod(200) (usuario criado com sucesso)

{"mensagem": "Usuario ou senha errados"} Cod(401) (usuario ou senha não corresponde ao banco)

DELETAR USUÁRIO:

API: <http://localhost/DesafioPubFuture/usuario/deletar/>

Método: POST

Necessário enviar: deletar: 1 //enviar POST deletar:1 por motivos de segurança mínima, além disso será deletada a conta que está logada, sem estar logado não é possível deletar uma conta

RESPOSTAS:

Cod(204) (usuario deletado com sucesso)

{"mensagem": " Erro ao deletar usuario "} Cod(400) (necessário POST deletar:1)

{"mensagem ": " Login Necessário "} cod(400)(necessário estar logado para poder deletar a própria conta)

LISTAR USUÁRIOS:

API: <http://localhost/DesafioPubFuture/usuario/listar/>

Ao acessar o link a lista de usuários será retornada, mas é necessário estar logado com um usuário

RESPOSTAS:

Cod(204) (usuario deletado com sucesso)

{"mensagem ": " Login Necessário "} cod(400)(necessário estar logado para ver a lista de usuários)

CONTA:

IMPORTANTE: A conta funciona relacionada a um usuário, só é possível criar a conta se estiver com um usuário logado para ser associada

CADASTRANDO CONTA:

API: <http://localhost/DesafioPubFuture/conta/cadastrar/>

Método: POST

Necessário enviar: nomeBanco, tipoDeConta, codConta

NomeBanco e tipoDeConta são enviados como códigos

Códigos abaixo estão disponíveis para uso:

```
$tipoDeConta = [  
    '1' => 'Conta Corrente',  
    '13' => 'Poupança'  
];  
  
$nomeBanco = [  
    '1' => 'Banco do Brasil S.A.',  
    '33' => 'Banco Santander (Brasil) S.A',  
    '104' => 'Caixa Econômica Federal',  
    '260' => 'nuBank',  
    '341' => 'Banco Itaú S.A.'  
];
```

RESPOSTAS:

```
{{  
    "mensagem": "OK",  
    "conta": {  
        "id": "3",  
        "saldo": "0",  
        "tipoDeConta": "13",  
        "nomeBanco": "1",
```

```
"userId": "16",
```

```
"codConta": "123456789"
```

```
}
```

```
}} Cod(200) (usuario criado com sucesso)
```

```
{"mensagem": "o código do tipo de conta está incorreto"} Cod(400) (código de conta não existe na lista acima)
```

```
{"mensagem": "o código do banco está incorreto"} cod(400) (Código do banco não existe na lista acima)
```

```
{"mensagem ": " Login Necessário "} cod(400)(necessário estar logado para adicionar conta a um usuário)
```

```
{"mensagem ": " campos obrigatórios faltando "} cod(400)(necessário enviar via post todos os campos acima para adicionar uma conta ao usuário)
```

```
{"mensagem ": " codConta já cadastrado no sistema"}cod(400)(não pode existir uma conta com mesmo código)
```

LISTAR CONTAS:

API: <http://localhost/DesafioPubFuture/conta/listar/>

Entrando no link pelo insomnia será disponibilizado as contas relacionadas ao usuário logado

RESPOSTAS:

```
{"mensagem ": " Login Necessário "} cod(400)(necessário estar logado para ver a lista de contas)
```


TRANSFERIR ENTRE CONTAS:

API: <http://localhost/DesafioPubFuture/conta/transferir/>

Método: POST

Necessário enviar: codContaPartida, tipoDeContaPartida(em código), codContaDestino, tipoDeContaDestino(em código), valorTransferencia

\$tipoDeConta = ['1' => 'Conta Corrente', '13' => 'Poupança'];

Segurança: o codContaPartida é verificada com o usuário logado, valor de transferência pode ser enviado com sinal ou sem sinal(+-), isto será corrigido pela API

RESPOSTAS:

{"mensagem ": "OK"} cod(200)(transferência realizada com sucesso)

{"mensagem ": " Login Necessário "} cod(400)(necessário estar logado para ver a lista de contas)

{"mensagem ": " codigo conta partida ou tipo de conta partida incorreto ou não pertence ao usuario logado "} cod(400)(verifique se o codContaPartida confere com o tipoDeContaPartida ou se a conta realmente pertence ao usuário logado)

{"mensagem ": " codigo conta destino ou tipo de conta destino incorreto"} cod(400)(verifique o codContaDestino e o tipoDeContaDestino)

DELETAR CONTA:

API: <http://localhost/DesafioPubFuture/conta/deletar/>

Método: POST

Necessário enviar: codConta

Segurança: o codConta é verificada com o usuário logado, valor de transferencia pode ser enviado com sinal ou sem sinal(+-), isto será corrigido pela API

RESPOSTAS:

Cod(200) (conta deletada com sucesso)

{"mensagem ": " Login Necessário "} cod(400)(necessário estar logado para ver a lista de contas)

{"mensagem ": " codido conta incorreto ou não pertence ao usuario logado "} cod(400)(necessário que codContaPartida pertença ao usuário logado)

{"mensagem ": " Erro ao deletar conta "} cod(400) (erro do servidor)

DESPESAS:

IMPORTANTE: As despesas estão ligadas a uma conta, é necessário que a conta exista e pertença ao usuário logado

CADASTRAR DESPESAS:

API: <http://localhost/DesafioPubFuture/despesas/cadastrar/>

Método: POST

Necessário enviar: categoria, codConta, dataEntrada, dataPrevista, descricao, valor

IMPORTANTE: Ao deixar dataEntrada em branco o valor já será debitado da conta automaticamente e a data salva como dia atual (não consegui tempo para fazer um sistema para que as despesas ou receitas caíam automaticamente quando chegar a data, então se for selecionado uma data a despesa/receita não será efetuada no saldo da conta)

Datas podem ser enviada 2022-01-15 ou 20220115 YYYYMMDD

Categoria será também enviada como código

Códigos abaixo estão disponíveis para uso:

CATEGORIA = [

'1' => 'conta água ou energia',

'2' => 'aluguel',

'3' => 'condominio',

'4' => 'internet',

'5' => 'telefone',

'6' => 'estudo',

'7' => 'alimentação',

'8' => 'cimpasto',

'9' => 'lazer',

'10' => 'transporte'

];

RESPOSTAS:

```
[
{
  "id": "64",
  "valor": "-1750",
  "dataPrevista": "2022-01-15 00:00:00",
  "dataEntrada": "2021-02-10 00:00:00",
  "descricao": "teste",
  "categoria": "1",
  "contald": "16",
  "codConta": "123456789",
  "ativo": "0"
}
```

] cod(200) (despesa adicionada com sucesso a conta escolhida e pertencente ao usuário logado)

{"mensagem": " Login Necessário "} cod(400)(necessário para cadastrar contas)

{"mensagem": "o código categoria da despesa está incorreto"} cod(400) (selecionar um código de categoria de despesa existente na lista acima)

{"mensagem": "código conta incorreto ou não pertence ao usuário logado"} cod(400) (código de conta não pode ser relacionado ao usuário logado ou não existe no sistema)

{"mensagem": " Erro do servidor "} cod(500) (erro interno do servidor)

DELETAR DESPESAS:

IMPORTANTE: mencionado no item acima, não tive tempo de fazer um sistema para as contas serem descontadas por data, sendo assim se o saldo por desconto da conta ele não será retornado para a conta ao ser deletada.

API: <http://localhost/DesafioPubFuture/despesas/deletar/>

Método: POST

Necessário enviar: categoria, codConta, idDespesa

RESPOSTAS:

Cod(204) (conta deletada com sucesso)

{"mensagem": "código conta incorreto ou não pertence ao usuário logado"} cod(400) (código de conta não pode ser relacionado ao usuário logado ou não existe no sistema)

{"mensagem": "despesa não pertence a conta selecionada "} cod(400) (o id da despesa está incorreto ou não se relaciona com a conta selecionada)

{"mensagem": " Erro ao deletar despesa "} cod(500) (erro interno do servidor)

LISTAR DESPESAS:

API: <http://localhost/DesafioPubFuture/despesas/listar/>

Método: POST

Necessário enviar: codConta

RESPOSTAS:

```
{"mensagem": " Login Necessário "} cod(400)(necessário para listar contas)
```

```
{"mensagem": "código conta incorreto ou não pertence ao usuário logado"} cod(400) (código de conta não pode ser relacionado ao usuário logado ou não existe no sistema, sendo assim as despesas não serão listadas)
```

```
[
```

```
{
```

```
"id": "--",
```

```
"categoria": "conta água ou energia",
```

```
"codConta": "123456789",
```

```
"dataEntrada": "2021-02-10 00:00:00",
```

```
"dataPrevista": "2022-01-15 00:00:00",
```

```
"descricao": "teste",
```

```
"valor": "-1750",
```

```
"ativo": "0"
```

```
},
```

IMPORTANTE: Ativo é um interno para controle do servidor para saber se a conta já foi descontada sobre saldo da conta, sendo assim futuramente é possível retirar o valor da despesa ao ser excluída ou inativada

FILTRAR DESPESAS POR DATA:

API: <http://localhost/DesafioPubFuture/despesas/filtrarPorData>

Método: POST

Necessário enviar: codConta, dataInicial, DataFinal

IMPORTANTE: Datas deve ser enviada neste formato: 20220115 YYYYMMDD

RESPOSTAS:

`{"mensagem ": " Login Necessário "} cod(400)(necessário para buscar despesas)`

`{"mensagem": "código conta incorreto ou não pertence ao usuário logado"} cod(400)` (código de conta não pode ser relacionado ao usuário logado ou não existe no sistema, sendo assim as despesas não serão listadas)

Códigos de despesas serão listados dessa mesma forma que ao utilizar `despesas/listar`

```
[
{
  "id": "--",
  "categoria": "conta água ou energia",
  "codConta": "123456789",
  "dataEntrada": "2021-02-10 00:00:00",
  "dataPrevista": "2022-01-15 00:00:00",
  "descricao": "teste",
  "valor": "-1750",
  "ativo": "0"
}, cod(200)
```

IMPORTANTE: Ativo é um interno para controle do servidor para saber se a conta já foi descontada sobre saldo da conta, sendo assim futuramente é possível retirar o valor da despesa ao ser excluída

FILTRAR DESPESAS POR TIPO:

API: <http://localhost/DesafioPubFuture/despesas/filtrarPorTipo>

Método: POST

Necessário enviar: codConta, categoria (tipo de despesa enviada por código mencionado no item “cadastrar Despesas”)

RESPOSTAS:

{"mensagem": " Login Necessário "} cod(400)(necessário para buscar despesas)

{"mensagem": "código conta incorreto ou não pertence ao usuário logado"} cod(400) (código de conta não pode ser relacionado ao usuário logado ou não existe no sistema, sendo assim as despesas não serão listadas)

Códigos de despesas serão listados dessa mesma forma que ao utilizar despesas/filtrarPorTipo

```
[  
{  
  "id": "--",  
  "categoria": "conta água ou energia",  
  "codConta": "123456789",  
  "dataEntrada": "2021-02-10 00:00:00",  
  "dataPrevista": "2022-01-15 00:00:00",  
  "descricao": "teste",  
  "valor": "-1750",  
  "ativo": "0"  
}, cod(200)
```

RECEITAS:

IMPORTANTE: As receitas estão ligadas a uma conta, é necessário que a conta exista e pertença ao usuário logado

CADASTRAR RECEITAS:

API: <http://localhost/DesafioPubFuture/receitas/cadastrar/>

Método: POST

Necessário enviar: categoria, codConta, dataEntrada, dataPrevista, descricao, valor

IMPORTANTE: Ao deixar dataEntrada em branco o valor já será debitado da conta automaticamente e a data salva como dia atual (não consegui tempo para fazer um sistema para que as despesas ou receitas caíam automaticamente quando chegar a data, então se for selecionado uma data a despesa/receita não será efetuada no saldo da conta)

Datas podem ser enviada 2022-01-15 ou 20220115 YYYYMMDD

Categoria será também enviada como código

Códigos abaixo estão disponíveis para uso:

CATEGORIA = [

'1' => 'salário',

'2' => 'investimento',

'3' => 'presente'

];

RESPOSTAS:

[

{

"id": "--",

"valor": "+1750",

"dataPrevista": "2022-01-15 00:00:00",

"dataEntrada": "2021-02-10 00:00:00",

"descricao": "teste",

"categoria": "1",

"contaId": "16",

"codConta": "123456789",

"ativo": "0"

}

] cod(200) (receita adicionada com sucesso a conta escolhida e pertencente ao usuário logado)

{"mensagem": " Login Necessário "} cod(400)(necessário para cadastrar contas)

{"mensagem": "o código categoria da receita está incorreto"} cod(400) (selecionar um código de categoria de despesa existente na lista acima)

{"mensagem": "código conta incorreto ou não pertence ao usuário logado"} cod(400) (código de conta não pode ser relacionado ao usuário logado ou não existe no sistema)

{"mensagem": " Erro do servidor "} cod(500) (erro interno do servidor)

DELETAR RECEITAS:

IMPORTANTE: mencionado no item acima, não tive tempo de fazer um sistema para as contas serem descontadas por data, sendo assim se o saldo por descontado da conta ele não será retornado para a conta ao ser deletada.

API: <http://localhost/DesafioPubFuture/despesas/deletar/>

Método: POST

Necessário enviar: categoria, codConta, idReceita

RESPOSTAS:

Cod(204) (conta deletada com sucesso)

{"mensagem": "código conta incorreto ou não pertence ao usuário logado"} cod(400) (código de conta não pode ser relacionado ao usuário logado ou não existe no sistema)

{"mensagem": " receita não pertence a conta selecionada "} cod(400) (o id da despesa está incorreto ou não se relaciona com a conta selecionada)

{"mensagem": " Erro ao deletar receita "} cod(500) (erro interno do servidor)

LISTAR RECEITAS:

API: <http://localhost/DesafioPubFuture/receitas/listar/>

Método: POST

Necessário enviar: codConta

RESPOSTAS:

{"mensagem": " Login Necessário "} cod(400)(necessário para listar contas)

{"mensagem": "código conta incorreto ou não pertence ao usuário logado"} cod(400) (código de conta não pode ser relacionado ao usuário logado ou não existe no sistema, sendo assim as despesas não serão listadas)

[

{

"id": "--",

"categoria": " salário ",

"codConta": "123456789",

"dataEntrada": "2021-02-10 00:00:00",

"dataPrevista": "2022-01-15 00:00:00",

"descricao": "teste",

"valor": "+1750",

"ativo": "0"

},

IMPORTANTE: Ativo é um interno para controle do servidor para saber se a conta já foi descontada sobre saldo da conta, sendo assim futuramente é possível retirar o valor da despesa ao ser excluída ou inativada

FILTRAR RECEITAS POR DATA:

API: <http://localhost/DesafioPubFuture/receitas/filtrarPorData>

Método: POST

Necessário enviar: codConta, dataInicial, DataFinal

IMPORTANTE: Datas deve ser enviada neste formato: 20220115 YYYYMMDD

RESPOSTAS:

{"mensagem ": " Login Necessário "} cod(400)(necessário para buscar despesas)

{"mensagem": "código conta incorreto ou não pertence ao usuário logado"} cod(400) (código de conta não pode ser relacionado ao usuário logado ou não existe no sistema, sendo assim as despesas não serão listadas)

Códigos de despesas serão listados dessa mesma forma que ao utilizar despesas/listar

```
[  
{  
  "id": "--",  
  "categoria": "salário",  
  "codConta": "123456789",  
  "dataEntrada": "2021-02-10 00:00:00",  
  "dataPrevista": "2022-01-15 00:00:00",  
  "descricao": "teste",  
  "valor": "+1750",  
  "ativo": "0"  
}, cod(200)
```

IMPORTANTE: Ativo é um interno para controle do servidor para saber se a conta já foi descontada sobre saldo da conta, sendo assim futuramente é possível retirar o valor da despesa ao ser excluída

FILTRAR RECEITAS POR TIPO:

API: <http://localhost/DesafioPubFuture/receitas/filtrarPorTipo>

Método: POST

Necessário enviar: codConta, categoria(tipo de receita)

RESPOSTAS:

Códigos de despesas serão listados dessa mesma forma que ao utilizar despesas/filtrarPorTipo

[

{

"id": "--",

"categoria": "salário",

"codConta": "123456789",

"dataEntrada": "2021-02-10 00:00:00",

"dataPrevista": "2022-01-15 00:00:00",

"descricao": "teste",

"valor": "+1750",

"ativo": "0"

}, cod(200)

{"mensagem": "o código categoria da receita está incorreto"} cod(400) (selecionar um código de categoria de despesa existente na lista de códigos categoria despesas)

{"mensagem": " Login Necessário "} cod(400)(necessário para buscar despesas)

{"mensagem": "código conta incorreto ou não pertence ao usuário logado"} cod(400) (código de conta não pode ser relacionado ao usuário logado ou não existe no sistema, sendo assim as despesas não serão listadas)