

DESAFIO W3



GERENCIADOR DE PROCESSOS
DE CARTÃO

SUMÁRIO:

- Requisitos da aplicação.
 - Estudo sobre DDD e camadas de dominio da aplicação.
 - Fluxograma de Operações do cartão.
 - Definição dos requisitos para um cliente válido.
 - Estrutura C4 Model.
 - Teste unitários das regras de negócio.
 - Documentação da API com swagger.
 - Documentação das respostas da API.
 - Demonstração de uma interface de fluxo de cartão.
 - Demonstração das entidades no banco de dados.
-

REQUISITOS DA APLICAÇÃO:

1. Solitação de Um Cartão: Solicitar um novo cartão com dados do cliente para verificação. Gerando um novo cartão com status inicial de 'solicitado'.
2. Ativação do Cartão: Ativar um cartão que esteja com o status de 'aprovado' para ser gerenciado.
3. Cadastro e Redefinição de senha: Cadastrar uma senha em um novo cartão ou redefinir a senha de um cartão já existente.
4. Bloqueio Temporário: Mediante um motivo válido, bloquear temporariamente o cartão impedido de realizar qualquer operação e recebendo o status 'BloqueadoTemporariamente'.
5. Comunicação de Perda ou Roubo: Conforme o tipo de ocorrência informado, bloquear imediatamente o cartão, recebendo o status 'Bloqueado_PerdaRoubo'.
6. Cancelamento Definitivo: Diante de um motivo de cancelamento, atribuir ao cartão o status 'Cancelado' e torna-lo permanentemente inutilizável.
7. Solicitação de Aumento de Limite: Mediante uma justificativa válida, o limite do cartão de crédito pode ser aumentado.
8. Histórico do Cartão (Extra!): Exibe o registro de todos os processos já realizados em um determinado cartão.

ESTUDO SOBRE DDD (DOMAIN-DRIVEN DESIGN)

O que é o padrão DDD?

Uma abordagem de desenvolvimento de software focada no domínio do negócio. Defende a colaboração entre especialistas do domínio e desenvolvedores para criar um modelo rico e alinhado com as regras de negócio do sistema.

Qual impacto real?

- Reduz falhas de comunicação: todos falam a mesma "linguagem ubíqua", facilitando entendimento entre áreas técnicas e de negócio.
- Aumenta a qualidade das decisões técnicas: o sistema é modelado com base em regras reais, não em suposições.
- Melhora a adaptabilidade do sistema: mudanças no negócio podem ser refletidas com clareza e segurança no código.

Conceitos-chave:

- Entidade — objeto com identidade
- Objeto de Valor — representa atributos sem identidade própria
- Serviço de Domínio — lógica que não pertence a uma entidade
- Repositório — abstrai o acesso aos dados
- Bounded Context — delimita fronteiras de significado e regras
- Linguagem Ubíqua — linguagem comum entre técnicos e especialistas

CAMADAS DE DOMÍNIO DA APLICAÇÃO:

SERVIÇO DE DOMÍNIO

Cartão

Cliente

BOUNDED CONTEXTS

Comunicação de perda ou roubo

Cancelamento definitivo

Solicitação de aumento de
limite

Ativação do cartão

Histórico do cartão

Bloqueio temporário

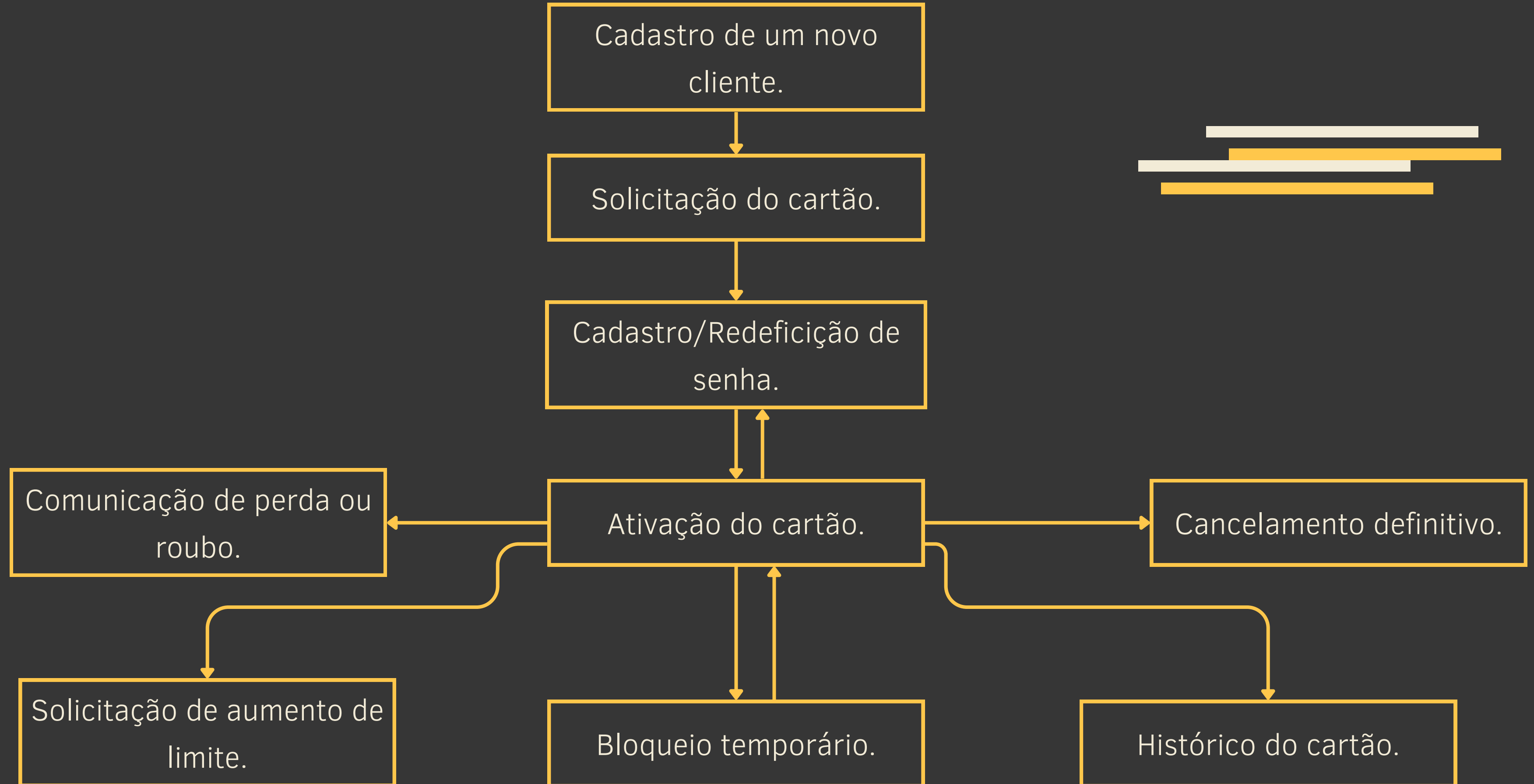
Cadastro/Redefinição de senha

Comunicação

Gerenciamento de Cartão

Segurança/Senha

FLUXOGRAMA DE OPERAÇÕES:



DEFINIÇÃO DOS REQUISITOS PARA UM CLIENTE VÁLIDO.

CPF válido:

- Deve conter exatamente 11 dígitos numéricos.
- Não pode ser composto por números repetidos (ex: "11111111111").
- Deve estar formatado corretamente (sem letras ou caracteres especiais além de pontos e traços, que são removidos internamente).
- Não pode estar duplicado no sistema (único por cliente).

Nome completo:

- Não pode ser nulo ou vazio.
- Deve ter no mínimo 2 caracteres.
- Deve conter apenas letras e espaços (sem números ou símbolos)

Data de nascimento:

- Não pode ser nula.
- O cliente deve ter pelo menos 18 anos de idade.

Renda mensal:

- Não pode ser nula.
- Não pode ser negativa.
- Acima de um salário mínimo (aprox: R\$ 1.518,00).

ESTRUTURA C4 MODEL:

This is a private board

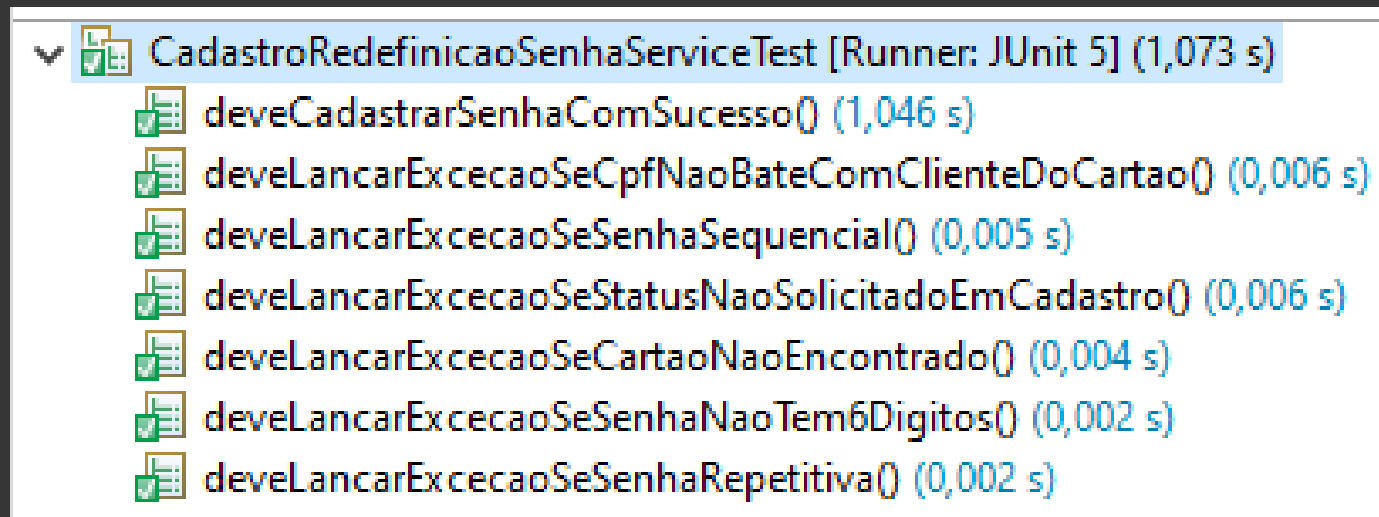
Only signed in users with permission can access the board.

Sign in

Sign up

TESTE UNITÁRIOS REALIZADOS:

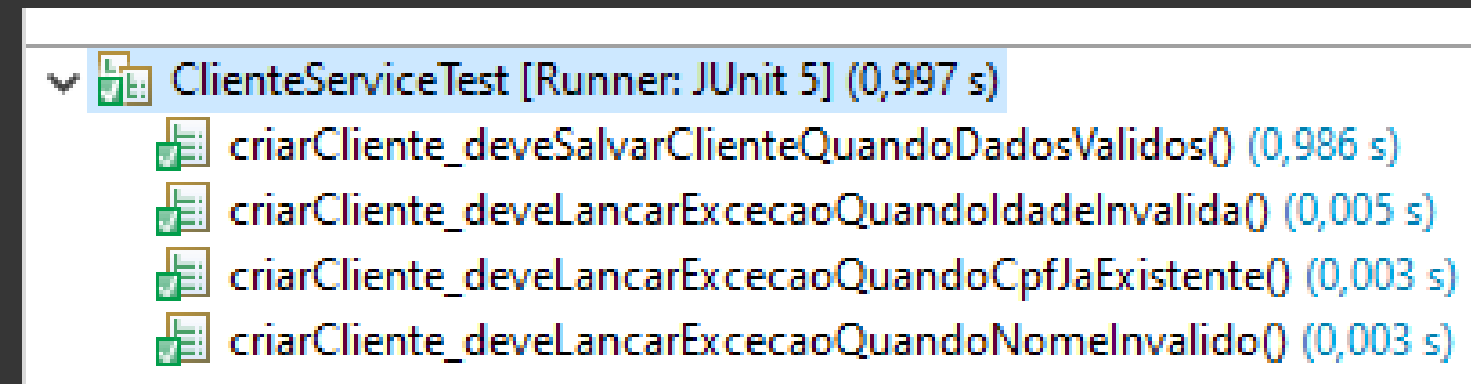
CADASTRO/REDEFINIÇÃO SENHA



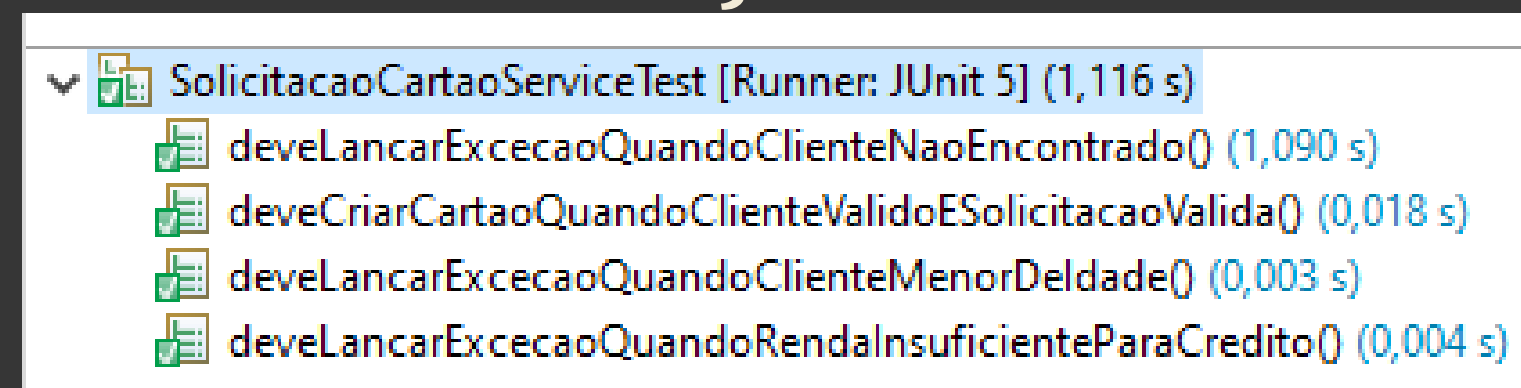
Teste do cadastro e a redefinição de senha do cartão (CadastroRedefinicaoSenhaServiceTest):

- Garante o cadastro de senha somente para cartões com status adequado.
- Valida a estrutura da senha (exatamente 6 dígitos, não sequencial, não repetitiva).
- Verifica se o CPF informado corresponde ao titular do cartão.
- Garante que a senha seja armazenada com segurança (hash) e que o status do cartão seja atualizado corretamente.

CLIENTE:



SOLICITAÇÃO CARTÃO:



Testes dos Serviços de Cartão e Cliente (SolicitacaoCartaoServiceTest e ClienteServiceTest):

- Garante a criação de cartão somente para clientes válidos que existem, são maiores de idade e possuem renda suficiente para o tipo de cartão solicitado.
- Impede solicitações de clientes inexistentes, menores ou com renda insuficiente.
- Verifica o cadastro de cliente com dados válidos, bloqueando CPFs duplicados.
- Valida o formato do nome do cliente e a idade mínima para cadastro.
- Utiliza Mockito para simular os repositórios e isolar as regras de negócio.
- Usa JUnit para validar a lógica de persistência e validação de forma precisa e segura.

DOCUMENTAÇÃO COM SWAGGER:

Gerenciador de processos de cartão v1 OAS 3.1

[/v3/api-docs](#)

Projeto de gerenciamento de ciclo completo de transações de cartões.

[Apache 2.0](#)

Servers

[http://localhost:8080 - Generated server url](#)

Solicitação Solicitação de um novo cartão

POST [/api/cartoes/solicitacoes](#) Cria um novo cartão no momento da solicitação sem uma senha cadastrada.

Limite Solicitação de aumento de limite do cartão

POST [/api/cartoes/limite](#) Aumenta o limite do cartão baseado na renda do usuário e no valor solicitado.

Cadastro e Redefinição Cadastro e redefinição de senha dos cartões

POST [/api/cartoes/senha/redefinir](#) Redefinição da senha de um cartão já ativo.

POST [/api/cartoes/senha/cadastrar](#) Cadastro de uma senha no cartão após solicitação.

Histórico Histórico do cartão

GET [/api/historico-cartao/{numeroCartao}](#) Retorna todas as operações já realizadas em um cartão ao longo do tempo.

Perda e Roubo Atualiza o status cartão em caso de perda ou roubo

DOCUMENTAÇÃO DAS REPOSTAS DA API:

Code	Description
------	-------------

200	Retorna o cliente
-----	-------------------

Media type

Controls Accept header.

[Example Value](#) | [Schema](#)

```
{
  "id": 0,
  "dataNascimento": "2025-06-29",
  "name": "string",
  "cpf": "string",
  "rendaMensal": 0.1,
  "dataCriacaoConta": "2025-06-29T16:07:56.698Z"
}
```

404	Cliente não encontrado
-----	------------------------

Media type

[Example Value](#) | [Schema](#)

```
{
  "id": 0,
  "dataNascimento": "2025-06-29",
  "name": "string",
  "cpf": "string",
  "rendaMensal": 0.1,
  "dataCriacaoConta": "2025-06-29T16:07:56.700Z"
}
```

500	Não existe um cliente com esse id
-----	-----------------------------------

Media type

[Example Value](#) | [Schema](#)

```
{
  "id": 0,
  "dataNascimento": "2025-06-29",
  "name": "string",
  "cpf": "string",
  "rendaMensal": 0.1,
  "dataCriacaoConta": "2025-06-29T16:07:56.702Z"
}
```

SUGESTÃO DE INTERFACE



Felipe Silva

 Editar seu perfil

Cliente: Felipe Viera Brazão e Silva

Saldo: R\$ 1.630,00



Solicitar cartão



Cadastrar senha



Ativar cartão



Trocar Conta

ESTRUTURA BANCO DE DADOS:

<https://console.neon.tech/app/projects/proud-surf-98940193/branches/br-mute-poetry-ac0t1kkh/tables?database=w3db>