

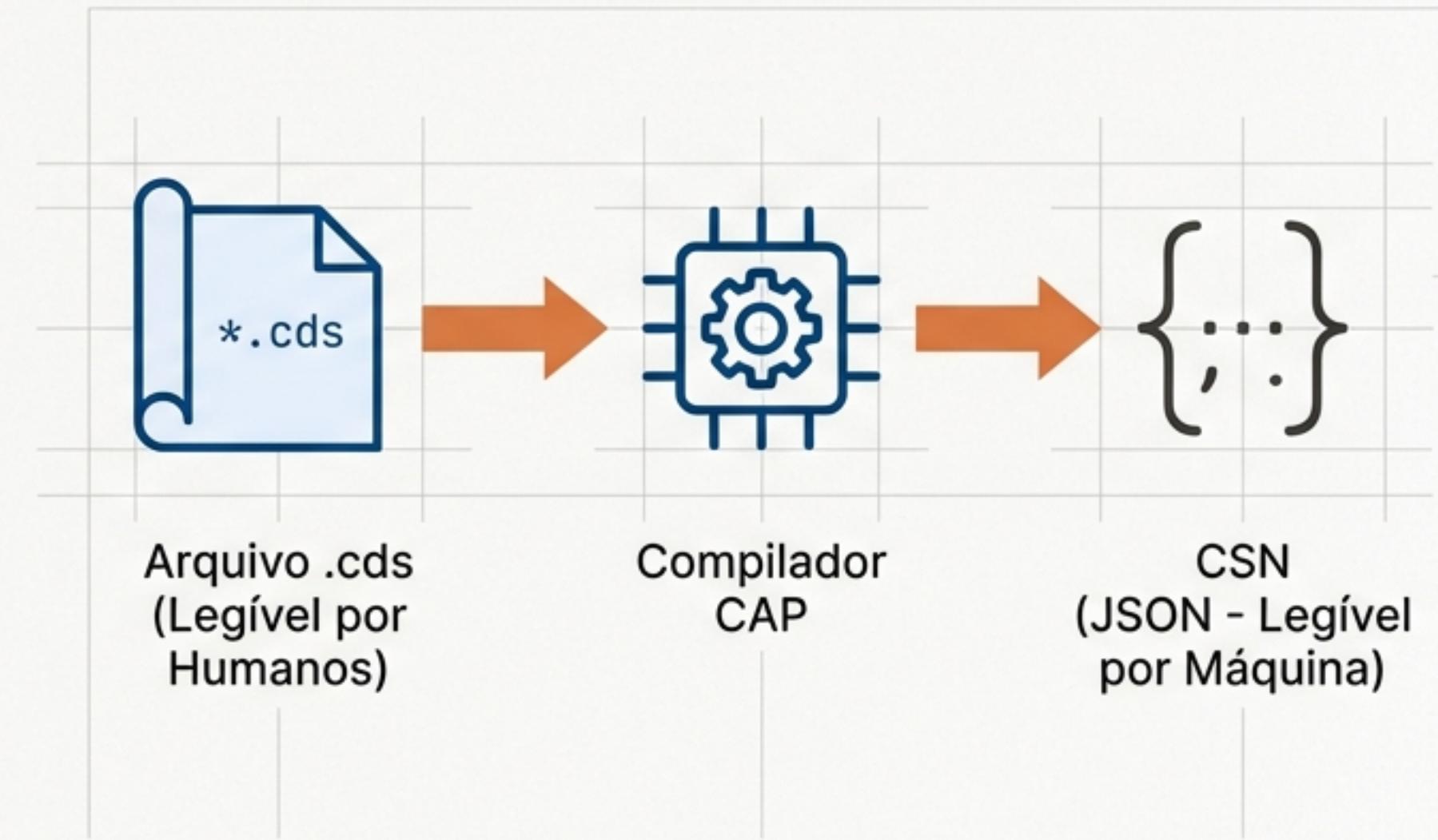
CDL: A Linguagem de Definição Conceitual

**Construindo Modelos de Dados e Serviços de
Forma Inteligente com SAP CAP**

O Que é a CDL? A Planta Baixa da Lógica de Negócio.

CDL é uma linguagem para definir todo o seu modelo de domínio — estruturas de dados, relacionamentos e APIs de serviço — em uma sintaxe coesa e elegante.

- + ***Legível por Humanos***: As definições são feitas em arquivos `*.cds`.
- + ***Compilada para Máquinas***: O compilador CAP transforma o código CDL em CSN (Core Schema Notation), o formato JSON que as máquinas entendem.
- + ***Abrangente****: Define tudo, desde o esquema do banco de dados até a camada de serviço OData.



A Fundação: Tipos Primitivos e Estruturas Básicas

A CDL fornece um rico conjunto de tipos de dados integrados que mapeiam diretamente para os tipos padrão de SQL e OData, garantindo interoperabilidade.

Tipo CDL	Mapeamento ANSI SQL
UUID	NVARCHAR(36)
Boolean	BOOLEAN
Integer	INTEGER
Decimal(p,s)	DECIMAL
String(l)	NVARCHAR
Date	DATE
Timestamp	TIMESTAMP

(1) Mapeamentos concretos para bancos de dados específicos podem variar.

O Coração do Modelo: Entidades e Tipos Customizados

Use a palavra-chave `entity` para definir dados estruturados com elementos tipados e chaves primárias. Crie tipos customizados com `type` para promover a reutilização.

```
// Entidade para Livros
entity Books {
    key ID : UUID;
    title : String(111);
    stock : Integer;
}
```

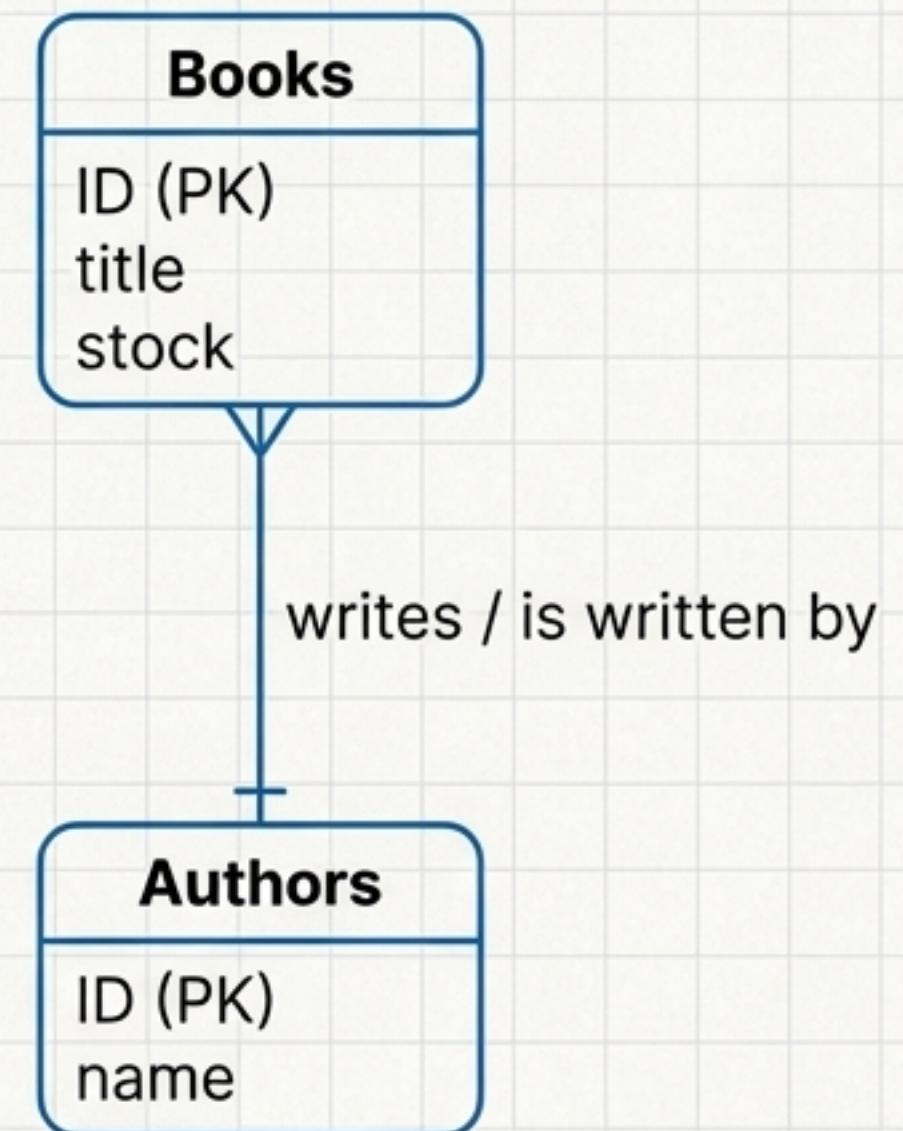
```
// Entidade para Autores
entity Authors {
    key ID : UUID;
    name : String;
}
```

Conectando os Pontos: Associações e Composições

‘Associações’ definem relacionamentos, permitindo que o CAP gerencie chaves estrangeiras automaticamente. ‘Composições’ definem uma relação de posse estrita (pai-filho).

```
// model.cds
entity Books {
    key ID : UUID;
    title : String(111);
    stock : Integer;
    author : Association to Authors; // <-- A mágica acontece aqui
}

entity Authors {
    key ID : UUID;
    name : String;
    books : Association to many Books on books.author = $self;
}
```



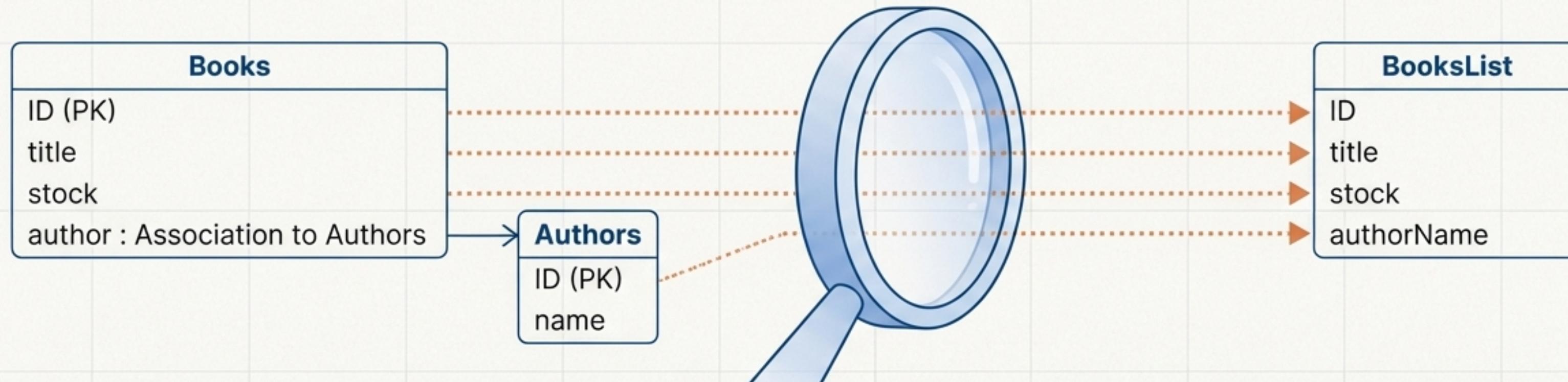
Elementos Inteligentes: Campos Virtuais e Calculados

A CDL permite definir elementos que não são persistidos (`virtual`) ou que são derivados de outros elementos (`calculated`), simplificando a lógica de negócio e as consultas.

```
entity Employees {  
    firstName : String;  
    lastName : String;  
  
    // Calculado em tempo de leitura (on-read)  
    name : String = firstName || ' ' || lastName; ←   
  
    // Calculado na escrita e persistido no banco (on-write)  
    name_stored : String = (firstName || ' ' || lastName) stored; ←   
}
```

Criando Novas Perspectivas: Views e Projeções

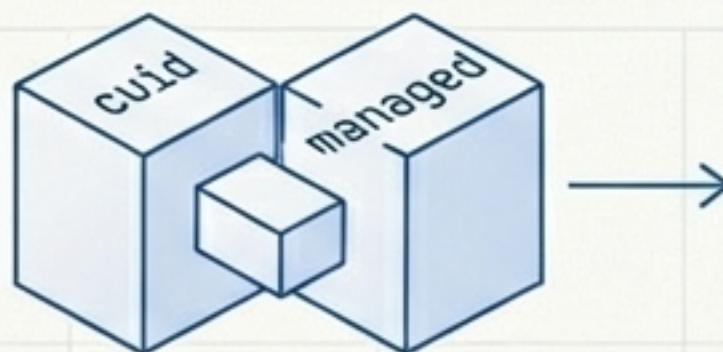
‘Views’ (‘as select from’) e ‘Projeções’ (‘as projection on’) permitem derivar novas entidades, de forma similar a views SQL, para criar conjuntos de dados específicos para diferentes casos de uso.



```
// A 'view' simplificada para uma lista de livros
entity BooksList as projection on Books {
    ID,
    title,
    stock,
    author.name as authorName
};
```

Acelere o Desenvolvimento: O Poder dos Aspects

‘Aspects’ são “mixins” reutilizáveis. Defina um conjunto de campos, anotações ou lógica uma única vez e aplique-o a múltiplas entidades, promovendo o princípio DRY (Don’t Repeat Yourself).



```
using { cuid, managed } from '@sap/cds/common';

// Nossas entidades agora herdam chaves UUID e campos
// de auditoria (createdAt, createdBy, modifiedAt, modifiedBy)
entity Books : cuid, managed {
    title      : String(111);
    stock      : Integer;
    author     : Association to Authors;
}
```

Adicionando Metadados e Comportamento: O Mundo das Anotações

Anotações (iniciadas com `@`) são a chave para enriquecer qualquer parte do seu modelo com metadados, desde rótulos para campos de UI até a especificação de comportamentos OData.

```
@cds.autoexpose // Expõe a entidade automaticamente no serviço
entity Books : cuid, managed {
    @title: 'Título do Livro'
    title      : String(111);

    @readonly
    stock      : Integer;

    author     : Association to Authors;
}
```

Expõe via OData API

Título do Livro

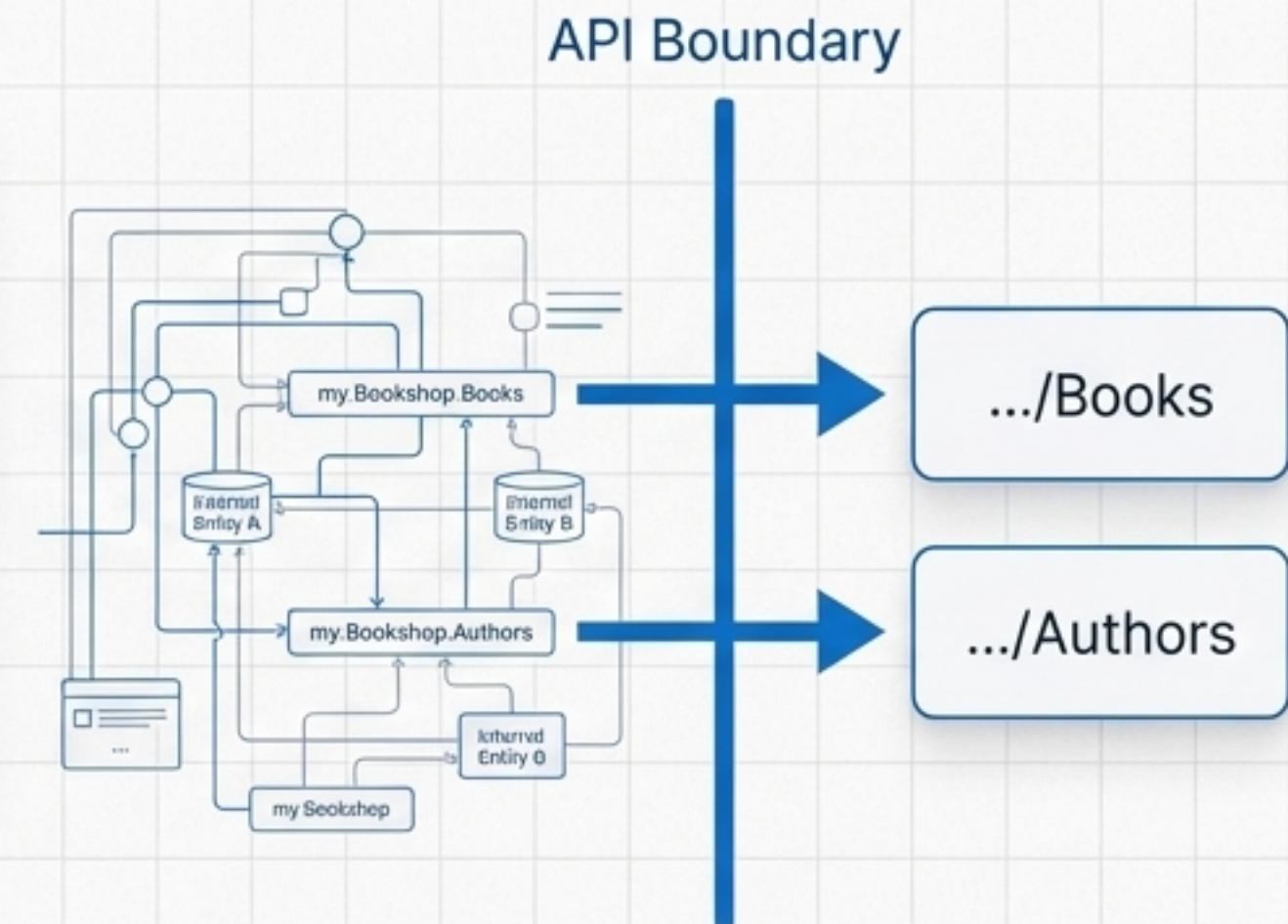


Expondo para o Mundo: Definições de Serviço

O bloco `service` define quais entidades do seu modelo são expostas como endpoints OData. O CAP cuida automaticamente de gerar e servir a API.

```
// A definição do nosso serviço de catálogo
service CatalogService {
    @readonly
    entity Books as projection on my.Bookshop.Books;

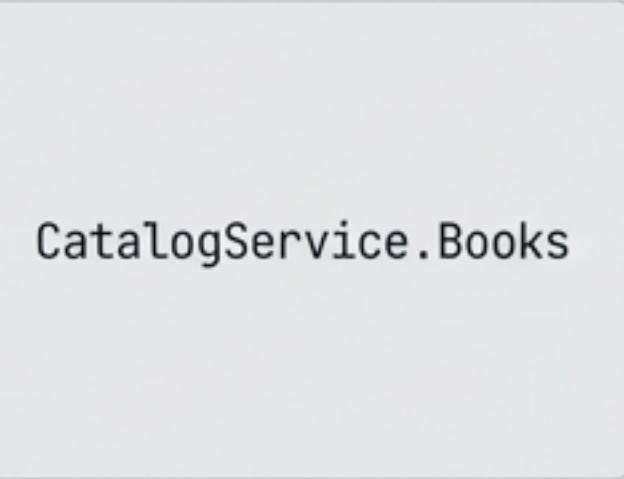
    @readonly
    entity Authors as projection on my.Bookshop.Authors;
}
```



Além do CRUD: Ações e Funções Customizadas

Defina `actions` (para operações que alteram o estado) e `functions` (para cálculos e leituras complexas) diretamente na sua definição de serviço para implementar lógicas de negócio específicas.

 action addRating →



```
// Estendendo nossa entidade de serviço com uma ação customizada
extend entity CatalogService.Books with actions {
    action addRating (stars: Integer);
}
```

Resumo da Jornada: Da Fundação ao Serviço Completo

A CDL é uma linguagem declarativa e poderosa que permite definir todo o domínio da sua aplicação em uma única planta baixa coesa e legível.



Key Benefits

Legibilidade: O modelo de domínio vira documentação.

Reutilização: Aspects e Tipos evitam repetição.

Poder: Uma definição gera o DB e a API.

Para um mergulho profundo e referência completa, consulte a documentação oficial do SAP CAP.

