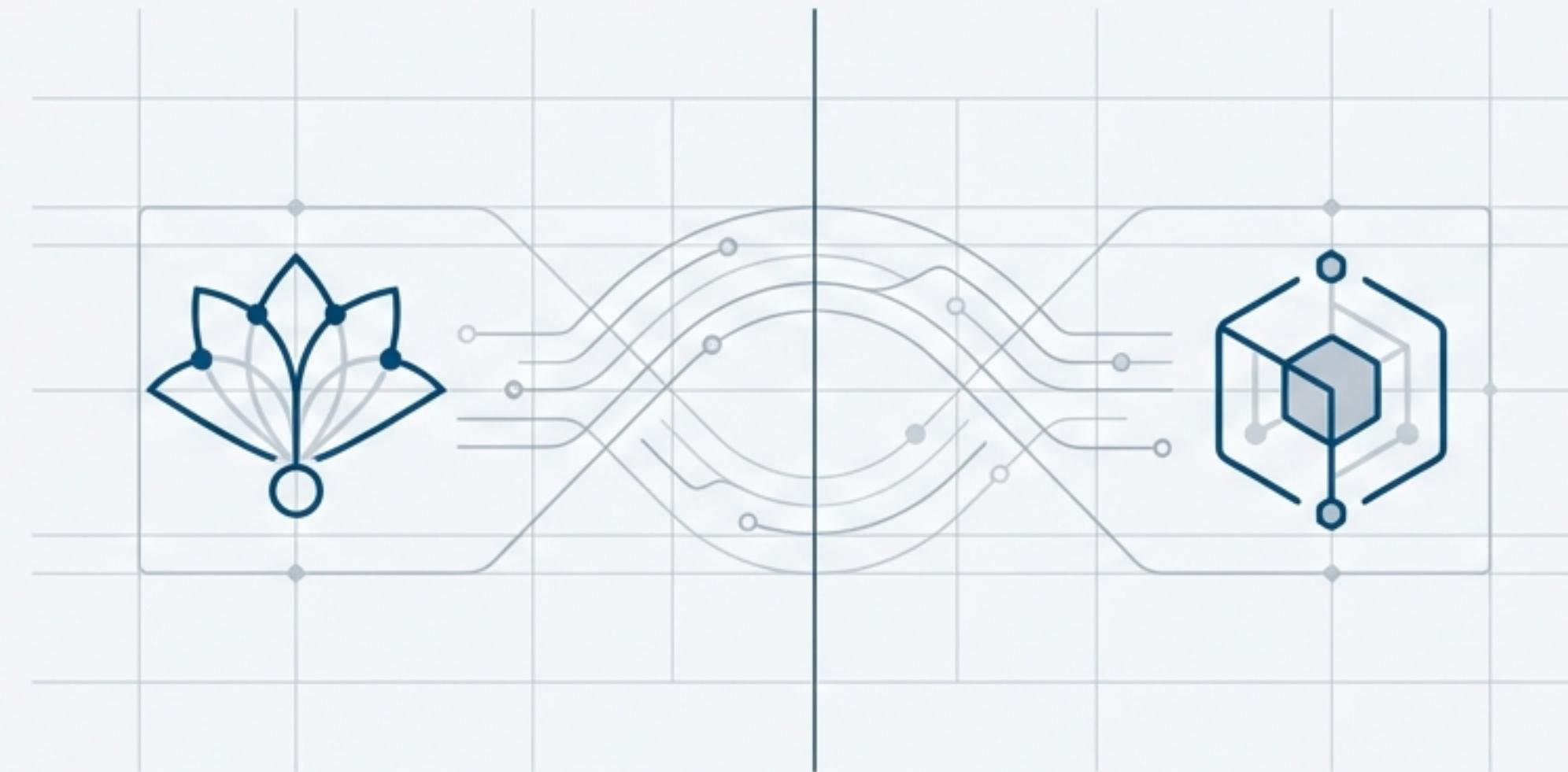


# Dominando UIs SAP Fiori com CAP



Um Guia para o Desenvolvimento  
Orientado a Metadados

# A Symbiose Perfeita: Backend Inteligente, Frontend Genérico



O CAP expõe dados e serviços. O Fiori Elements renderiza a UI. As Anotações OData são a linguagem comum que permite que eles conversem de forma inteligente, descrevendo a semântica dos dados para a interface.

# A Jornada do Desenvolvedor: Da Definição à UI em 4 Passos

1



Definir o Serviço

```
service CatalogService {  
}
```

2



Adicionar o App  
Fiori

Estrutura `app/browse/`

3



Anotar a UI

```
annotate CatalogService.Books  
with @UI...
```

4



Visualizar  
Instantaneamente

SAP Fiori Preview

Este fluxo de trabalho integrado permite um ciclo de desenvolvimento rápido, onde as mudanças no modelo de UI são refletidas em tempo real.

# O Contrato da UI: O Poder das Anotações

## O que são?

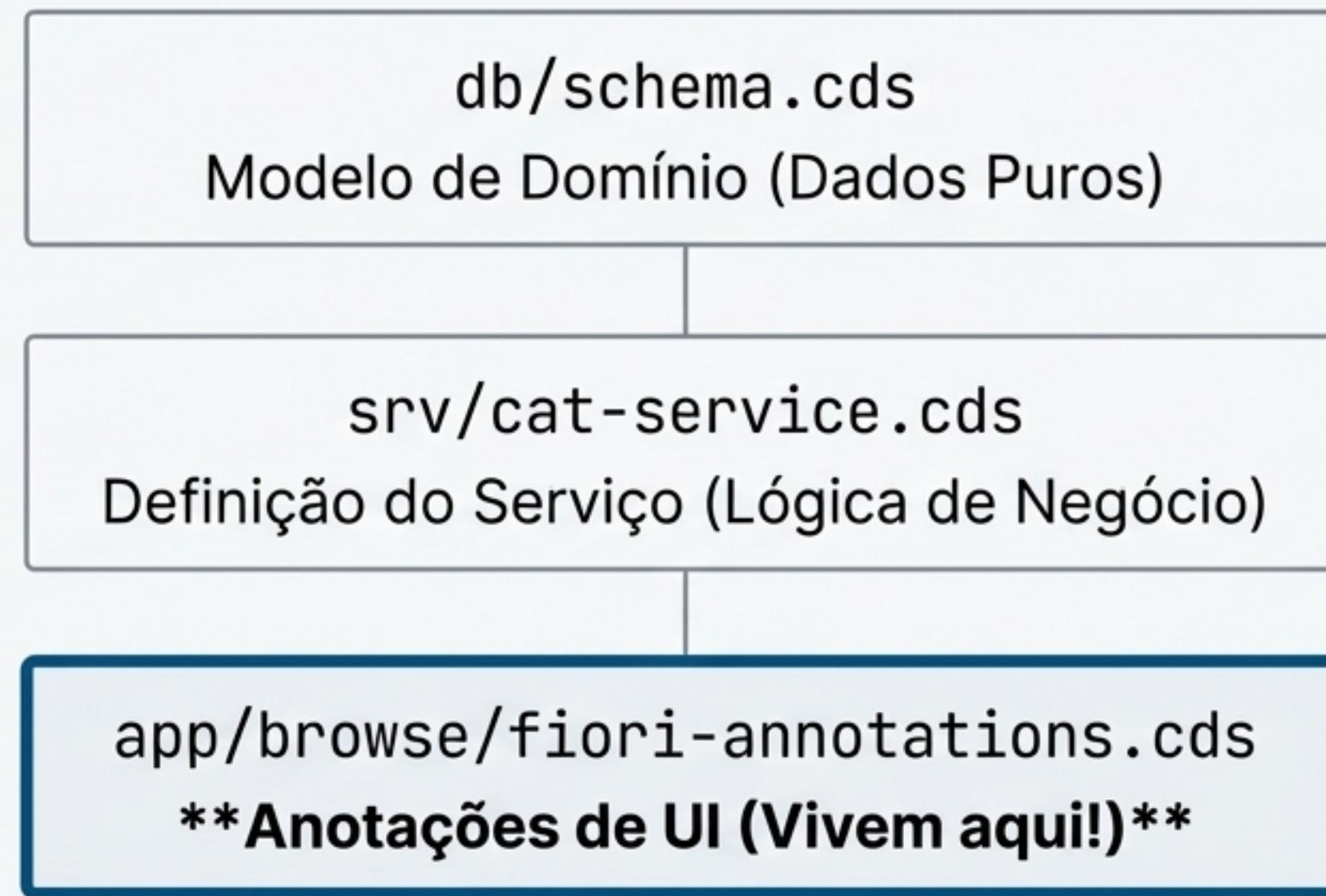
Instruções semânticas que descrevem o que os dados significam e como devem ser apresentados, não apenas quais são os dados. Elas formam um contrato entre o backend e a UI.

```
annotate CatalogService.Books with @(
    UI: {
        LineItem: [
            {Value: title}, -----
            {Value: author.name} -----
        ]
    } } );
```

## O Efeito

→ Title	→ Name
Book One	Author A
Book Two	Author B
Book Three	Author C
Book Four	Author D
Book Five	Author E
Book Five	Author F

# Arquitetura Limpa: Separação de Interesses



Modelos de domínio e serviços permanecem agnósticos à UI. As anotações vivem junto com a aplicação que as consome. Isso resulta em código mais limpo, reutilizável e fácil de manter.

# Anotações em Ação (1/2): Construindo a Lista de Relatórios

```
cds
annotate CatalogService.Books with @(
    UI: {
        SelectionFields: [ price, currency_code ],
        LineItem: [
            {Value: title},
            {Value: author.name, Label:'Autor'},
            {Value: price, Label:'Preço'},
            {Value: currency.symbol, Label:''}
        ]
    }
);
```

The screenshot shows a SAP Fiori List Report interface. At the top, there is a search bar with fields for 'Preço' (Price) and 'Moeda' (Currency), and a 'Go' button. Below the search bar is a table with the following data:

Title	Autor	Preço	Moeda
The Hitchhiker's Guide to the Galaxy	Douglas Adams	42.00	€
1984	George Orwell	19.48	£
Brave New World	Aldeus Huxley	25.50	\$

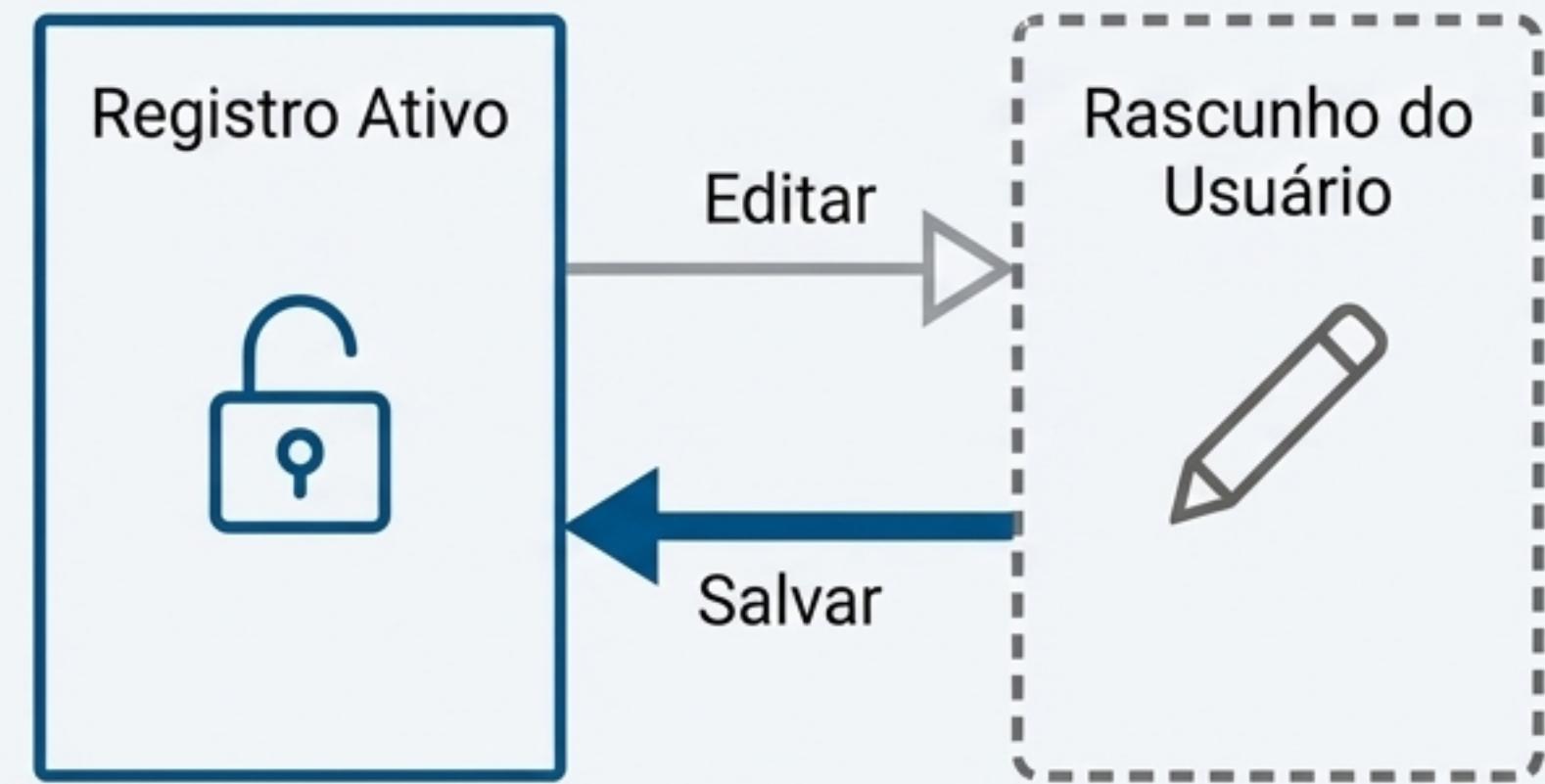
## Anotações em Ação (2/2): Detalhando a Página de Objeto

```
annotate CatalogService.Books with @(
    UI: {
        HeaderInfo: {
            TypeName: 'Livro',
            TypeNamePlural: 'Livros',
            Title: { Value: title }
        },
        Facets: [
            { $Type: 'UI.ReferenceFacet', Label: 'Detalhes',
              Target: '@UI.FieldGroup#Details' }
        ],
        FieldGroup #Details: { Data: [
            {Value: author.name}, {Value: genre.name},
            {Value: stock}
        ]}
    }
);
```



# Edição Segura e Contínua: O Conceito de Rascunho (Draft)

Permite que os usuários editem dados sem bloquear o registro ativo para outros. As alterações são salvas automaticamente em um rascunho, permitindo que o trabalho seja interrompido e retomado a qualquer momento, em qualquer dispositivo, sem perda de dados.



# A Mágica do CAP: Ativando o Draft com Uma Anotação

```
| annotate AdminService.Books with @odata.draft.enabled;
```

É isso. Adicionar esta anotação ao seu serviço instrui o framework CAP a gerenciar todo o ciclo de vida do rascunho: criação de tabelas de sombra, lógica de edição, ativação e exclusão. Toda a complexidade é abstraída.

Nota: Para entidades com dados localizados, a anotação `@fiori.draft.enabled` também é necessária na entidade base para garantir chaves primárias técnicas nas tabelas de texto.

# A Coreografia do Draft em Ação

The screenshot shows the SAP Fiori Book application. The title is "Wuthering Heights" by Emily Brontë. The user is in "Edit" mode, indicated by a pencil icon. They are changing the German translation from "Sturmhalde" to "Summtihüe". The application has tabs for Header, General, Translations, Details, and Admin. The "Translations" tab is active. It lists translations for various locales: German (De), English (En), French (Fr), and Spanish (Es). The "Details" section shows stock (12), price (11.11), and currency (GBP). A "Save" button is at the bottom.



1. Usuário clica em "Edit"

POST .../Books

→ Resposta: Cria o Rascunho.



2. Usuário digita em um campo

PATCH .../Books(ID=..., IsActiveEntity=false)

→ Resposta: Atualiza o Rascunho.



3. Usuário clica em "Save"

POST .../Books(ID=..., IsActiveEntity=false)/draftActivate

→ Resposta: Ativa o Rascunho, que se torna a nova entidade ativa.

# Guiando o Usuário: Ajudas de Valor (Value Helps)

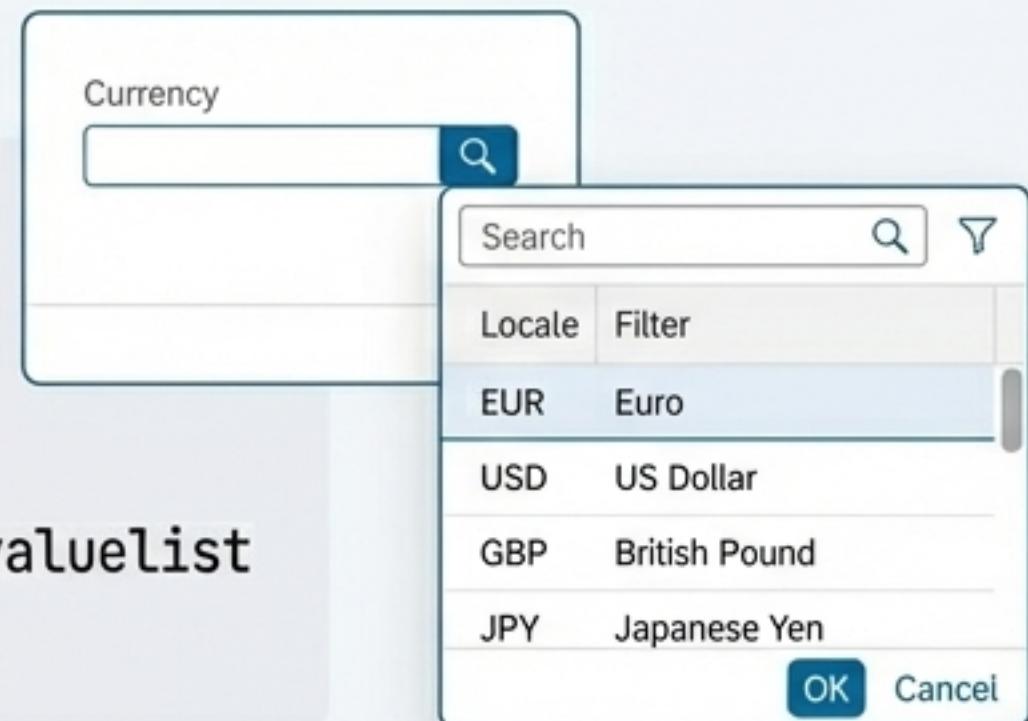
## The Challenge

**O Desafio:** Como garantir que os usuários insiram dados válidos e existentes (como moedas, países ou categorias) sem que precisem memorizar IDs ou códigos?

## The Solution

```
using { Currency } from '@sap/cds/common';

entity Books {
  ...
  currency : Currency; // A entidade Currency já possui @cds.odata.valuelist
}
```



A anotação `@cds.odata.valuelist` em uma entidade-alvo (como `Currencies` de `@sap/cds/common`) instrui a UI a renderizar um diálogo de seleção automaticamente para todas as associações que apontam para ela.

# Tornando a UI Interativa com Ações de Negócio

Aesthetic: Clean, modern, architectural, and editorial, inspired by Swiss design principles...

Definindo a Ação no Backend (srv/service.cds)

```
entity Travel actions {
    action acceptTravel() returns Travel;
};
```

Expondo a Ação na UI (app/annotations.cds)

```
annotate TravelService.Travel with @UI : {
    LineItem : [
        { $Type: 'UI.DataFieldForAction',
          Action: 'TravelService.acceptTravel',
          Label: 'Aceitar Viagem' }
    ]
};
```

<input type="checkbox"/>	Travel ID	Customer	Status	Total Price	
<input type="checkbox"/>	12345	Acme Corp	Pending	2450.00 EUR	<b>Aceitar Viagem</b>

# UI Inteligente: Controlando a Visibilidade com Base em Regras

## O Caso de Uso

O botão 'Aceitar Viagem' só deve estar ativo se o status da viagem for 'Aberto' ou 'Pendente'.

## The Declarative Solution (Code)

```
annotate TravelService.Travel with actions {
    acceptTravel @(
        Core.OperationAvailable: TravelStatus_code <> 'A' // 'A' = Approved
    )
};
```

Status: Aberto

Travel ID	Customer	Status	Total Price	
12345	Acme Corp	Aberto	2450.00 EUR	<button>Aceitar Viagem</button>

Status: Aprovado

Travel ID	Customer	Status	Total Price	
12345	Acme Corp	Aprovado	2450.00 EUR	<button>Aceitar Viagem</button>

# Tópico Avançado: Visualizando Dados em Árvore Hierárquica

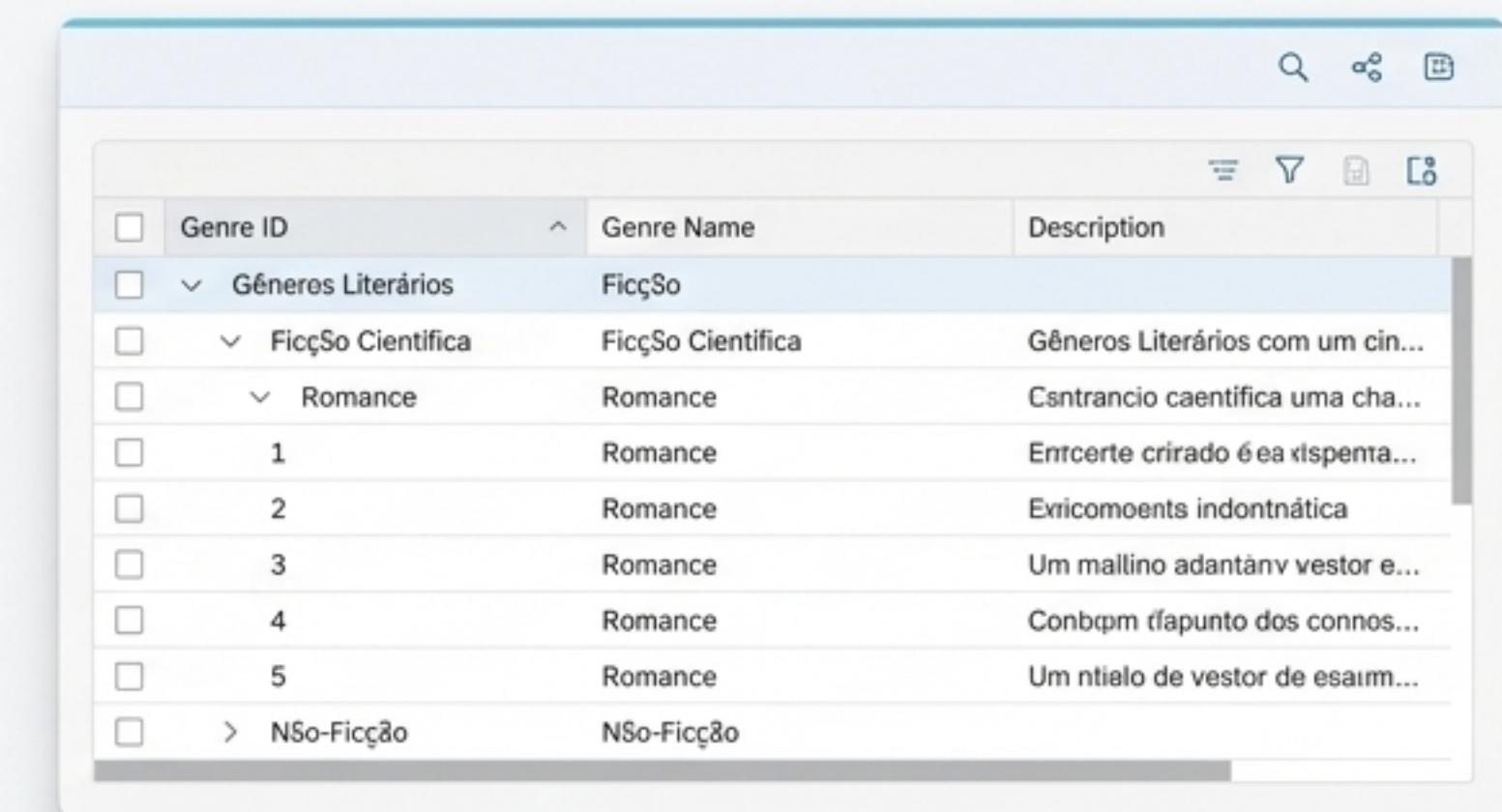
O Fiori Elements pode renderizar dados auto-referenciados (pai-filho) como uma tabela em **árvore interativa**. Isso é configurado através de uma combinação de **anotações no serviço** e **configurações** no `manifest.json` da UI.

## Step 1: Anotação no Serviço (CDS)

```
annotate AdminService.Genres
  with @Aggregation.RecursiveHierarchy #GenresHierarchy {
    NodeProperty: ID,
    ParentNavigationProperty: parent
};
```

## Step 2: Configuração da UI (manifest.json)

```
"controlConfiguration": {
  "@com.sap.vocabularies.UI.v1.LineItem": {
    "tableSettings": {
      "type": "TreeTable",
      "hierarchyQualifier": "GenresHierarchy"
    }
  }
}
```



The screenshot shows a Fiori application interface with a tree table control. The table has three columns: 'Genre ID', 'Genre Name', and 'Description'. The data is organized hierarchically:

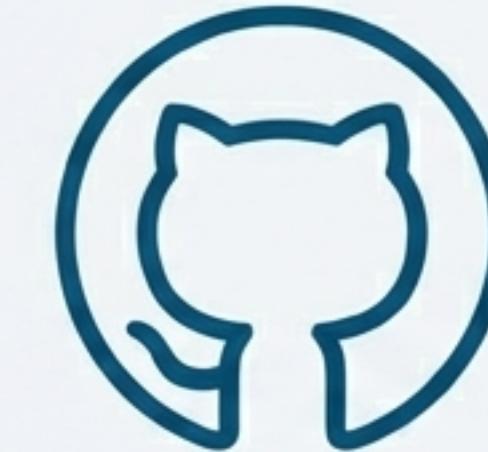
Genre ID	Genre Name	Description
1	Ficção	Gêneros Literários com um cin...
2	Romance	Conteúdo intelectual com um cin...
3	Romance	Conteúdo intelectual com um cin...
4	Romance	Conteúdo intelectual com um cin...
5	Romance	Conteúdo intelectual com um cin...
>	Não-Ficção	Conteúdo intelectual com um cin...

# Princípios Fundamentais e Próximos Passos

## Principais Lições

- ✓ **Foco em Metadados:** Descreva a UI com anotações semânticas, não com código imperativo.
- ✓ **Separação de Interesses:** Mantenha os modelos de domínio e serviço limpos e agnósticos à UI.
- ✓ **Poder Declarativo:** Ative funcionalidades complexas como *Drafts* e *Value Helps* com anotações simples.
- ✓ **Produtividade Máxima:** Utilize o Fiori Preview e as ferramentas de linguagem para acelerar drasticamente o ciclo de desenvolvimento.

## Comece a Construir Hoje



Explore os exemplos completos no repositório [capire/bookshop](#).

Clone, execute e modifique para solidificar seu aprendizado e iniciar seu próximo projeto Fiori com CAP.