

# Dominando a Privacidade de Dados com SAP CAP

Um guia prático para desenvolvedores sobre automação de conformidade na BTP

Este guia é um 'cookbook' visual e autônomo, projetado para ser lido e compartilhado. Ele mostra como o SAP Cloud Application Programming Model (CAP) simplifica a implementação de recursos de privacidade de dados.

# O Desafio da Conformidade: Respondendo às Perguntas do Usuário



## Direito de Acesso

“Quais dados sobre mim vocês têm armazenados?”

Obrigação Legal: Direito de Acesso (Ex: GDPR Art. 15)



## Transparência

“Quando meus dados pessoais foram armazenados ou alterados?”

Obrigação Legal: Transparência (Ex: GDPR Art. 15(1))



## Direito ao Esquecimento

“Por favor, apaguem todos os meus dados pessoais!”

Obrigação Legal: Direito ao Esquecimento

# A Solução CAP: Automação Inteligente na BTP

Para

“Quais dados sobre mim vocês têm?”



## Solução: SAP Personal Data Manager (PDM)

Permite que administradores informem indivíduos sobre os dados armazenados a respeito deles.

“Quando meus dados foram alterados?”



## Solução: SAP Audit Log Service

Armazena todos os logs de auditoria em um repositório comum e compatível, permitindo que auditores pesquisem e recuperem os registros.

“Apaguem meus dados!”



## Solução: SAP Data Retention Manager

Gerencia regras de retenção e residência para bloquear ou destruir dados pessoais. (Nota: Em desenvolvimento no CAP).

**Mensagem Central:** O CAP e a BTP possuem as ferramentas para automatizar as respostas.

# A Chave Mestra: Anotações `@PersonalData`



**Tudo começa aqui.** A primeira e, frequentemente, única tarefa do desenvolvedor é identificar entidades e elementos que contêm dados pessoais usando anotações @PersonalData.

Essas anotações no seu modelo de dados CDS são a base para que o CAP possa automatizar o log de auditoria, o gerenciamento de dados pessoais e a retenção de dados.

# Passo 1: Classificando Entidades com `@PersonalData.EntitySemantics`

## DataSubject

A entidade que descreve o titular dos dados (pessoa física identificada ou identificável).

```
annotate my.Customers with @PersonalData: { EntitySemantics:  
  'DataSubject' };
```

## DataSubjectDetails

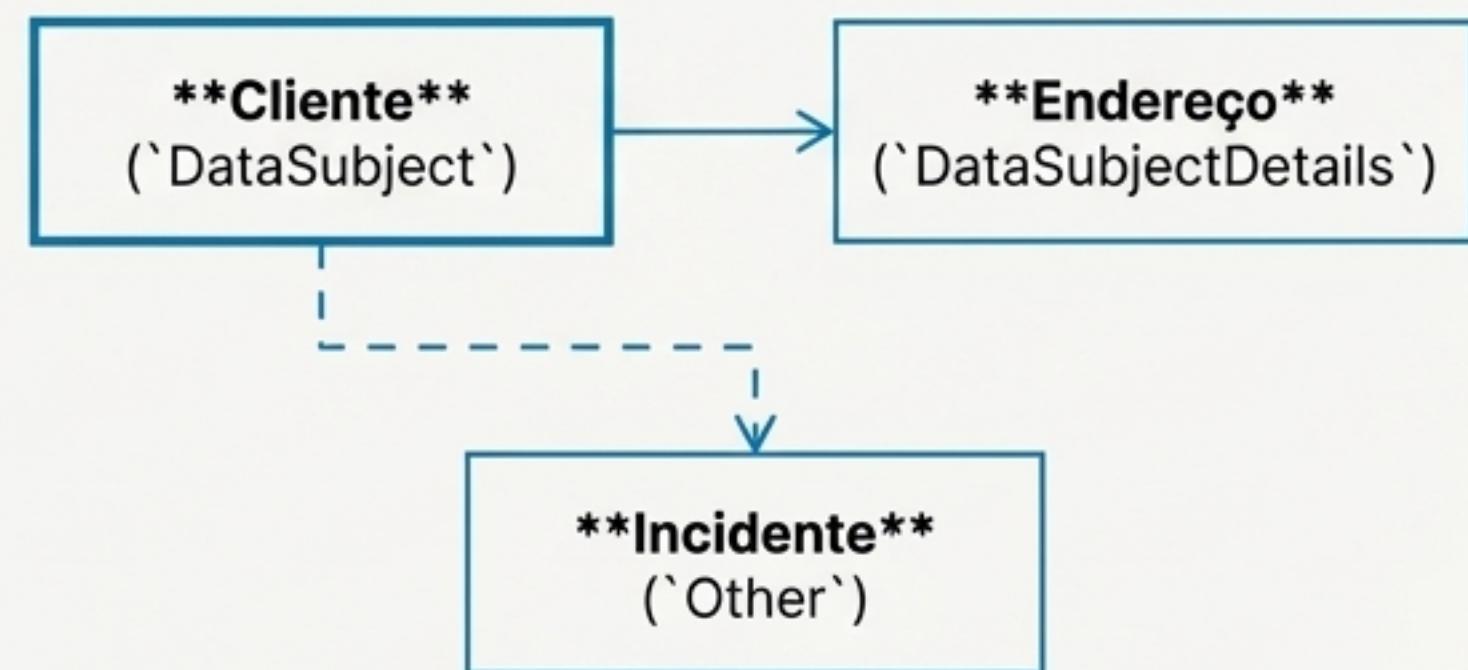
Entidades que contêm detalhes do titular dos dados, mas não o identificam por si sós.

```
annotate my.Addresses with @PersonalData: { EntitySemantics:  
  'DataSubjectDetails' };
```

## Other

Entidades com dados pessoais ou referências a titulares, como dados transacionais. Relevante para o log de auditoria.

```
cds annotate my.Incidents with @PersonalData: {  
  EntitySemantics: 'Other' };
```



## Passo 2: Vinculando Dados com `@PersonalData.FieldSemantics: "DataSubjectID"

**Conceito Chave:** Cada entidade anotada com `@PersonalData` precisa identificar um elemento `DataSubjectID`. Este é o elo que conecta todos os dados a um indivíduo.

**Para entidades** `DataSubject`  
Geralmente é a chave primária.

```
annotate my.Customers with {  
    ID @PersonalData.FieldSemantics: 'DataSubjectID';  
};
```

**Para** `DataSubjectDetails` **ou** `Other`  
Geralmente é uma associação (chave estrangeira) para o `DataSubject`.

```
annotate my.Incidents with {  
    customer @PersonalData.FieldSemantics: 'DataSubjectID'  
};
```



**Boa Prática:** Campos marcados como `DataSubjectID` devem ser `not null` para garantir que um valor esteja sempre presente.

# Passo 3: Marcando Campos Pessoais e Sensíveis

## `@PersonalData.IsPotentiallyPersonal`

**Propósito:** Marca campos que são pessoais. Modificações nestes campos exigem logs de auditoria.

Exemplo: `firstName`, `lastName`, `email`

## @PersonalData.IsPotentiallySensitive`

**Propósito:** Marca campos que são sensíveis. O **acesso** a estes campos exige logs de auditoria.

Exemplo: `creditCardNo`

```
cds
annotate my.Customers with @PersonalData: { ... } {
    ID          @PersonalData.FieldSemantics: 'DataSubjectID';
    firstName   @PersonalData.IsPotentiallyPersonal;
    lastName    @PersonalData.IsPotentiallyPersonal;
    email       @PersonalData.IsPotentiallyPersonal;
    phone       @PersonalData.IsPotentiallyPersonal;
    phone       @PersonalData.IsPotentiallyPersonal;
    dateOfBirth @PersonalData.IsPotentiallyPersonal;
    creditCardNo @PersonalData.IsPotentiallySensitive;
};
```

# Recompensa 1: Log de Auditoria Ativado em Um Comando

## Habilitar o Plugin

```
$ npm add @cap-js/audit-logging
```

Isso é tudo que você precisa para começar a registrar eventos relacionados a dados pessoais automaticamente.

## Ver em Ação (Localmente)

### Requisição `PATCH`

```
PATCH http://localhost:4004/admin/Customers(...)  
{  
  "firstName": "Jane",  
  "lastName": "Doe"  
}
```



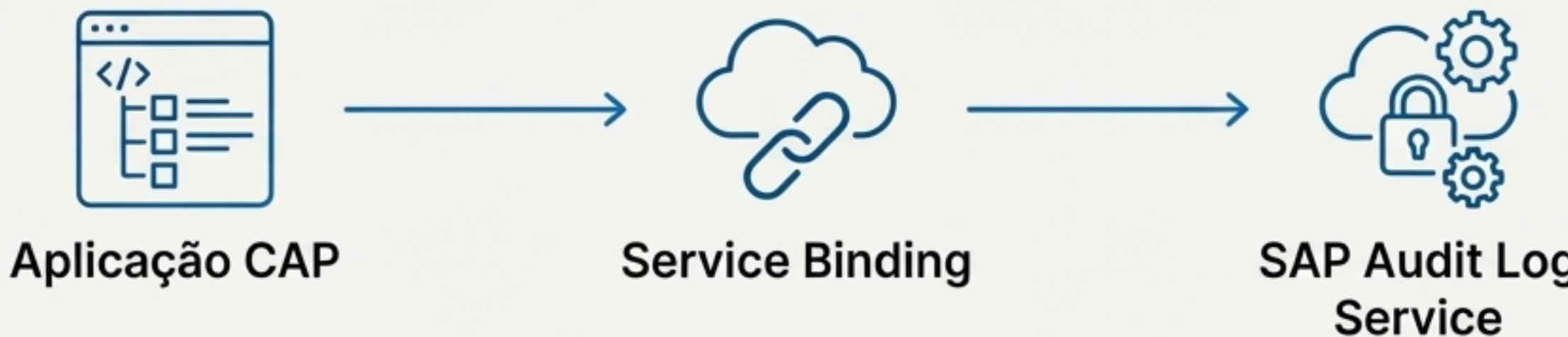
### Log Gerado Automaticamente

```
{  
  "data_subject": { "role": "Customer", ... },  
  "attributes": [  
    { "name": "firstName", "old": "Sunny", "new": "Jane" },  
    { "name": "lastName", "old": "Sunshine", "new": "Doe" }  

```

# Do Local à Produção: Integrando com SAP Audit Log Service

Para usar o serviço de log de auditoria da BTP em produção, são necessários alguns passos de configuração.



## Passos de Configuração Principais

- Criar Instância de Serviço:** No seu space BTP, crie uma instância do serviço `auditlog` com o plano `premium`.
- Configurar `mta.yml`:** Adicione a instância como um recurso existente (`org.cloudfoundry.existing-service`).
- Vincular o Serviço:** Na seção `requires` da sua aplicação no `mta.yml`, vincule o serviço de log.

## Como Acessar os Logs

- Opção 1:** Via API REST (usando o serviço `auditlog-management`).
- Opção 2:** Via UI (usando o `SAP Audit Log Viewer`).

# Indo Além do Automático: A API Programática de Auditoria

O serviço de log de auditoria do CAP é um serviço padrão, permitindo consumo programático para registrar eventos customizados.

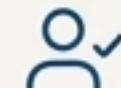
## Passo 1: Conectar ao Serviço

```
const audit = await cds.connect.to('audit-log');
```

## Passo 2: Enviar Mensagem de Log

```
await audit.log('PersonalDataModified', {
  data_subject: { type: '...', id: { ... }, role: '...' },
  object: { type: '...', id: { ... } },
  attributes: [
    { name: 'emailAddress', old: 'foo@example.com',
      new: 'bar@example.com' }
  ]
});
```

## Tipos de Eventos Pré-definidos

-  **SensitiveDataRead**: Para leitura de dados sensíveis.
-  **PersonalDataModified**: Para modificações em dados pessoais.
-  **ConfigurationModified**: Para mudanças de configuração.
-  **SecurityEvent**: Para eventos de segurança (ex: acesso não autorizado).

**\*\*Mensagem Chave:\*\*** Tenha controle total e extensibilidade para registrar qualquer evento de negócio relevante.

# Recompensa 2: Construindo a Interface para o SAP Personal Data Manager

O PDM precisa de um endpoint OData para buscar os dados pessoais. A melhor prática é criar um serviço CAP dedicado para isso.

## Passo 1: Criar Projeções Achatadas (Views)

O PDM precisa de estruturas achatadas. Para entidades com composições (ex: 'Incidents' e 'Conversations'), crie uma view.

```
entity IncidentConversationView as
  select from Incidents { ...,
    key conversation.ID as conversation_ID,
    conversation.message as
      conversation_message,
    customer.ID as customer_ID
  };
```

## Passo 2: Anotar as Views

A nova view também precisa de anotações '@PersonalData'.

```
annotate PDMService.IncidentConversationView with @(
  PersonalData.EntitySemantics: 'Other'
) {
  customer_ID @(
    PersonalData.FieldSemantics: 'DataSubjectID'
  );
}
```

## Passo 3: Anotar Campos de Contato

Anote os campos de busca na entidade 'DataSubject' com '@Communication.Contact' para que o PDM saiba como pesquisar.

```
annotate Customers with @(
  Communication.Contact: {
    n: { surname: lastName, given: firstName },
    email: [{ address: email }]
  });
});
```

# Protegendo o Acesso e Configurando a Segurança do PDM



## Passo 1: Restringir o Acesso ao Serviço

Use a anotação `@requires` no seu serviço `PDMService` para garantir que apenas usuários autorizados possam acessá-lo.

```
@requires: 'PersonalDataManagerUser'  
service PDMService @({path: '/pdm'}) { ... }
```



## Passo 2: Definir a Permissão no `xs-security.json`

A cláusula `grant-as-authority-to-apps` permite que o serviço PDM chame sua aplicação com a role `PersonalDataManagerUser`.

```
"scopes": [{  
    "name": "$XSAPPNAME.PersonalDataManagerUser",  
    "description": "Authority for Personal Data Manager",  
    "grant-as-authority-to-apps": ["$XSSERVICENAME(pdm)"]  
}]
```



## Passo 3: Instalar Dependência de Segurança

Para que a autenticação funcione, adicione o pacote `@sap/xssec`.

```
npm install @sap/xssec
```

# Conectando os Pontos: O Processo de Binding na BTP

## Fluxo de Configuração



## 1. DEPLOY

Faça o build (`cds build --production`)  
deploy (`cf create-service-push`) da sua  
aplicação CAP.



## 2. SUBSCRIBE

Inscreva-se no serviço 'Personal Data Manager' no BTP Cockpit.



### 3. CREATE INSTANCE

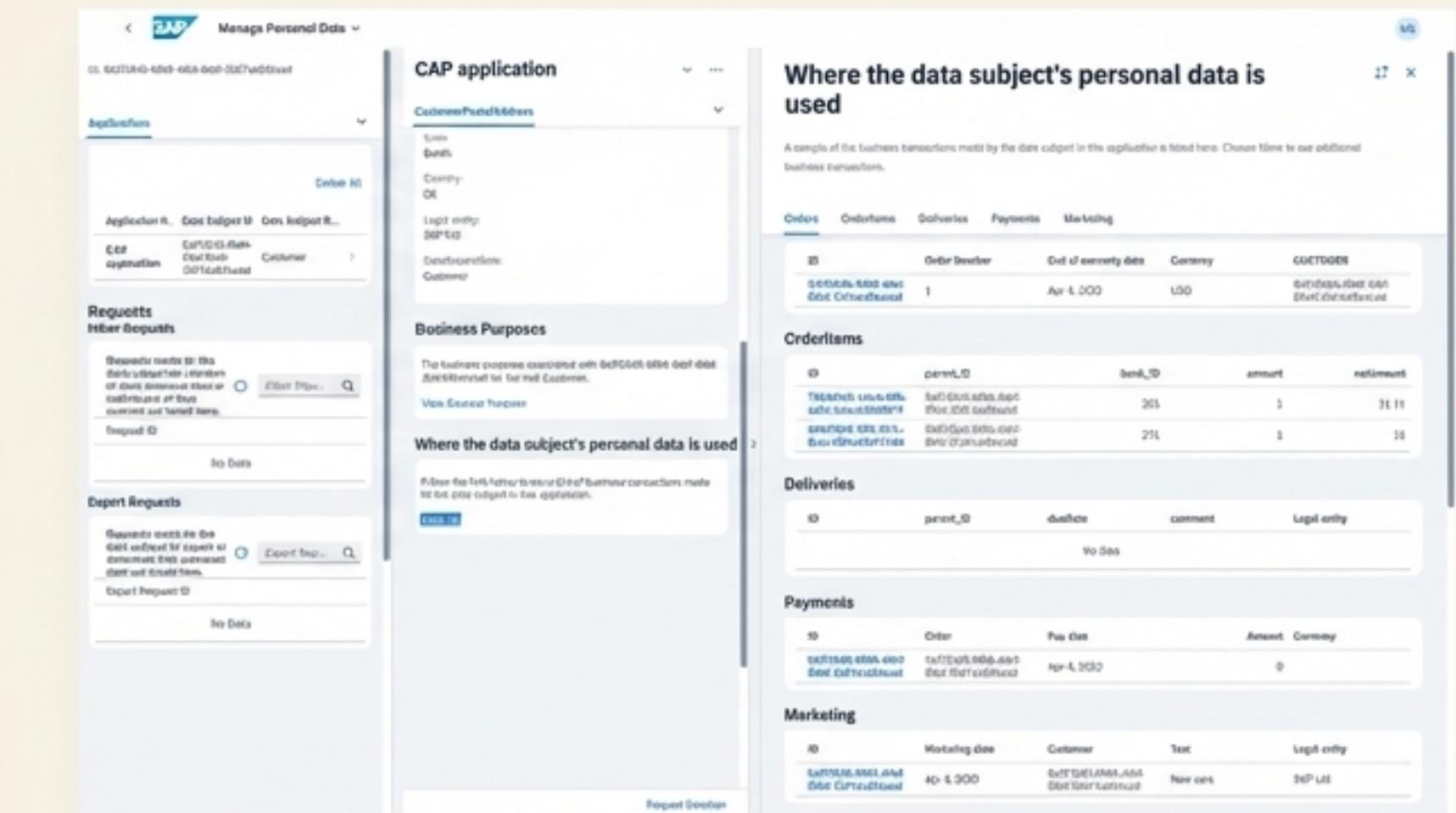
Crie uma instância do serviço PDM



#### **4. BIND**

Vincule a instânciado PDM à sua aplicação CAP implantada.

## O Resultado Final



Após a configuração, o PDM pode buscar e exibir os dados pessoais do titular dos dados diretamente da sua aplicação, de forma segura e em conformidade.

# Resumo da Jornada: Da Anotação à Conformidade

## 1. ANOTE



Classifique suas entidades e campos com anotações `@PersonalData`. Esta é a sua única tarefa fundamental.

## 2. HABILITE



Adicione os pacotes (@cap-js/audit-logging) e vincule os serviços da BTP (Audit Log, PDM) à sua aplicação.

## 3. CONFORME



O CAP e os serviços da BTP automatizam a geração de logs e a exposição de dados para o PDM.

**Mensagem Principal:** Com anotações declarativas no seu modelo de dados, o CAP automatiza tarefas complexas de privacidade de dados, liberando você para focar na lógica de negócio.

# Recursos e Próximos Passos

## Documentação Oficial CAP (CAPire)

[LINK: [cap.cloud.sap/docs/guides/data-privacy](https://cap.cloud.sap/docs/guides/data-privacy)]

## Guias de Referência Detalhados

[LINK: [cap.cloud.sap/docs/guides/data-privacy/annotations](https://cap.cloud.sap/docs/guides/data-privacy/annotations)]

[LINK: [cap.cloud.sap/docs/guides/data-privacy/audit-logging](https://cap.cloud.sap/docs/guides/data-privacy/audit-logging)]

[LINK: [cap.cloud.sap/docs/guides/data-privacy/personal-data-management](https://cap.cloud.sap/docs/guides/data-privacy/personal-data-management)]

## Aplicação de Exemplo (Incidents Management)

[LINK: [github.com/sap-samples/cap-incidents-management](https://github.com/sap-samples/cap-incidents-management)]



Comece a anotar seus modelos de dados hoje e transforme a conformidade de um fardo em um recurso automatizado.