

Dominando o Ciclo de Vida do CAP Java: Versões e Dependências

Um guia para criar aplicações estáveis, seguras e atualizadas.

A Disciplina do Versionamento Semântico



MAJOR

2.0.0

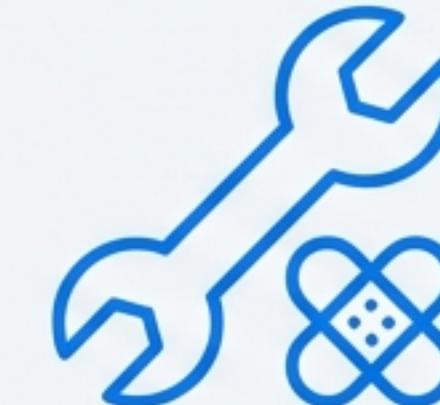
Introduz **mudanças incompatíveis**. Grandes evoluções anunciadas com antecedência.



MINOR

2.7.0

Adiciona **novas funcionalidades** de forma **compatível**. Este é o seu caminho principal de atualização.

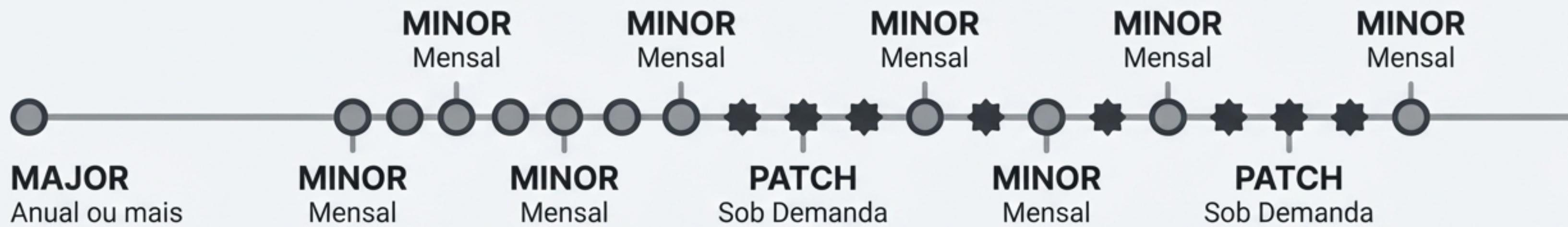


PATCH

2.7.1

Correções de bugs críticos e patches de segurança. Não contém novas features.

O Ritmo das Entregas: O que esperar e quando



Recomendamos fortemente consumir a última versão minor mensalmente para manter os esforços de migração futuros os menores possíveis.

Linha Ativa vs. Manutenção: Escolha sua estratégia de evolução



Linha Ativa

Recebe: Atualizações MINOR + PATCH

Onde as novas features acontecem. O caminho para inovação contínua.

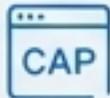


Linha de Manutenção

Recebe: Apenas atualizações PATCH

Segurança e estabilidade para aplicações que necessitam de um horizonte maior para migrar para uma nova versão major.

Checklist de Dependências: Linha Ativa (Versão 4.x)

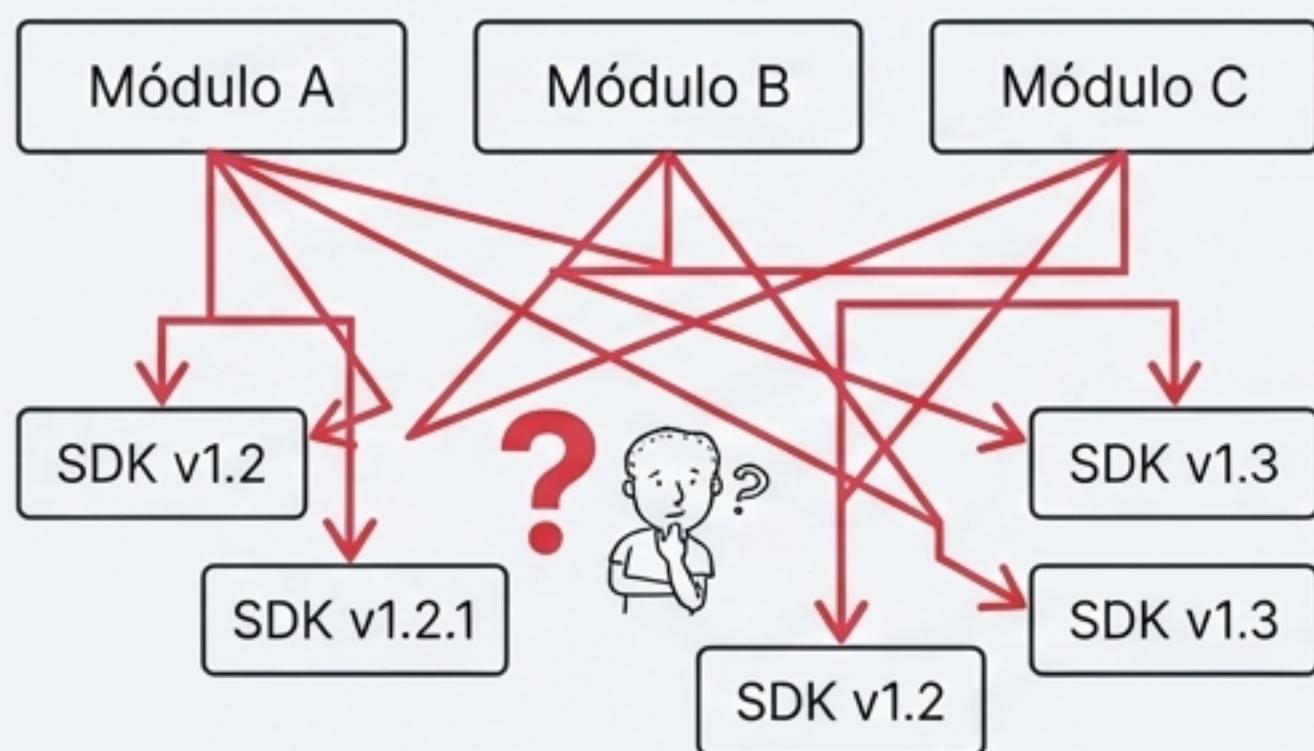
Dependência	Versão Mínima	Versão Recomendada
 CAP Java	n/a	latest 4.x
 JDK	17	21
 Maven	3.8.8	3.9.9
 @sap/cds-dk	8	latest
 @sap/cds-compiler	5	latest
 Spring Boot	3.0	latest 3.x
 XSUAA	3.1	latest
 SAP Cloud SDK	5.9	latest
 Java Logging	3.7	latest
 Node.js	20	22

Checklist de Dependências: Linha de Manutenção (Versão 3.10.x)

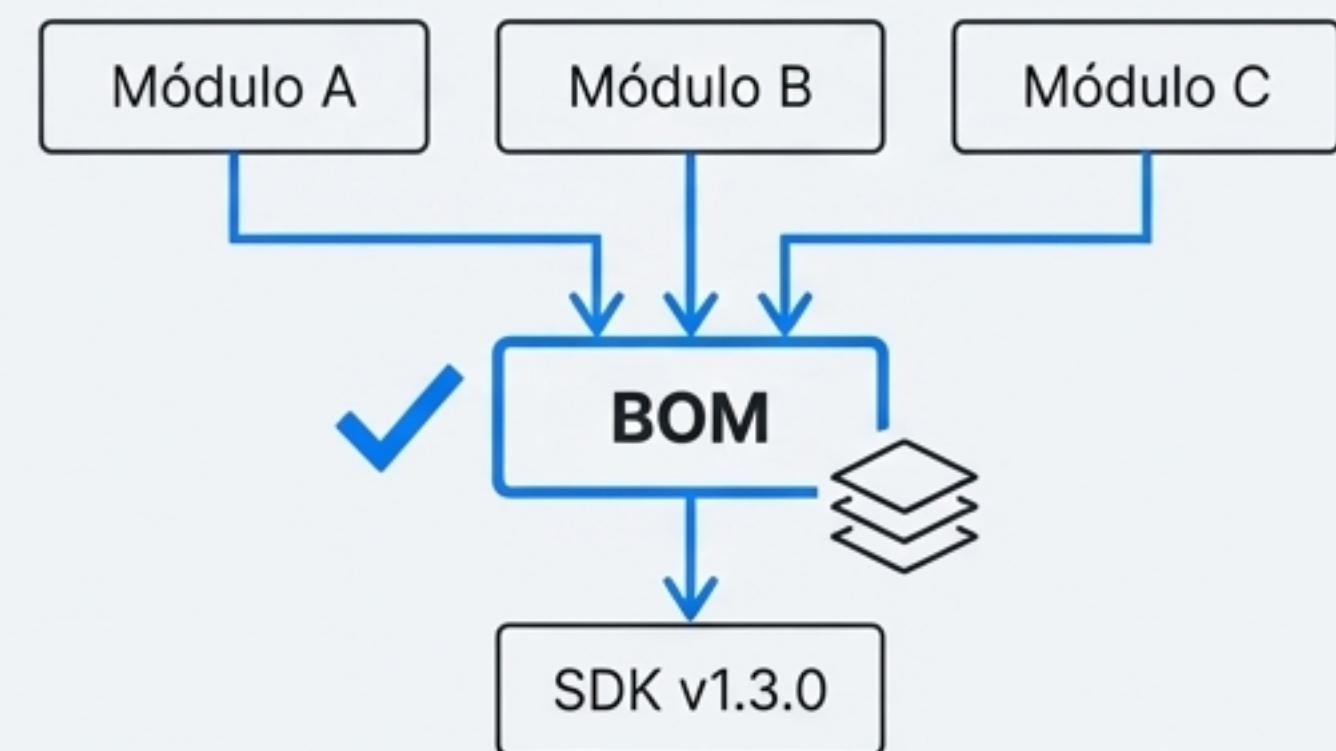
Dependência	Versão Mínima	Versão Recomendada
 CAP Java	n/a	latest 3.10.x
 JDK	17	21
 Maven	3.8.8	3.9.9
 @sap/cds-dk	7	latest 8.x
 @sap/cds-compiler	4	latest 5.x
 Spring Boot	3.0	latest 3.x
 XSUAA	3.0	latest
 SAP Cloud SDK	5.9	latest
 Java Logging	3.7	latest
 Node.js	18	20

A Regra de Ouro: Garanta a consistência com Bill of Materials (BOM)

Sem BOM



Com BOM



Misturar diferentes versões de artefatos de um mesmo SDK frequentemente resulta em erros de compilação ou problemas imprevisíveis em tempo de execução.

Use o Bill of Materials (BOM) para importar e gerenciar as versões de um SDK de forma centralizada e consistente.

Blueprint do `pom.xml`: Implementando os BOMs

```
<properties>
    <cds.services.version>[CAP Java version here]</cds.services.version>
    <cloud.sdk.version>[Cloud SDK version here]</cloud.sdk.version>
    <xsuua.version>[XSUAA version here]</xsuua.version>
</properties>

<dependencyManagement>
    <dependencies>
        <!-- CDS SERVICES -->
        <dependency>
            <groupId>com.sap.cds</groupId>
            <artifactId>cds-services-bom</artifactId>
            <version>${cds.services.version}</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
        <!-- CLOUD SDK -->
        <dependency>
            <groupId>com.sap.cloud.sdk</groupId>
            <artifactId>sdk-modules-bom</artifactId>
            <version>${cloud.sdk.version}</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
        <!-- SAP SECURITY -->
        <dependency>
            <groupId>com.sap.cloud.security</groupId>
            <artifactId>java-bom</artifactId>
            <version>${xsuua.version}</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
```

O Inevitável: Uma vulnerabilidade é descoberta em uma dependência



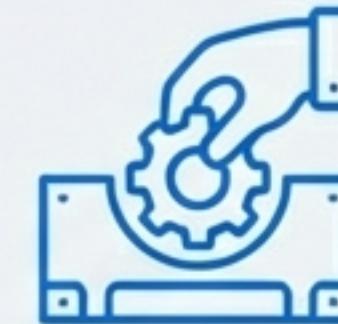
Uma vulnerabilidade em uma dependência (FOSS) pode ser publicada entre os releases do CAP Java, bloqueando o deploy da sua aplicação devido a falhas em scans de segurança.

Suas Opções Estratégicas: Aguardar ou Agir



Aguardar

- **Ação:** Aguardar o próximo release mensal do CAP Java com as dependências corrigidas.
- **Quando usar:** Ideal para vulnerabilidades não-críticas ou quando o cronograma permite.



Agir

- **Ação:** Especificar manualmente uma versão segura da dependência vulnerável.
- **Quando usar:** Essencial para vulnerabilidades críticas que bloqueiam o release.

A Intervenção Precisa: Sobrescrevendo a versão vulnerável

Adicione a versão segura da dependência **no início da seção <dependencyManagement>** do `pom.xml` principal da sua aplicação.

```
<dependencyManagement>
    <!-- Adicione a sua dependência corrigida aqui -->
    <dependency>
        <groupId>[group ID da dependência]</groupId>
        <artifactId>[artifact ID da dependência]</artifactId>
        <version>[versão segura aqui]</version>
    </dependency>
    <!-- O restante dos seus BOMs e dependências -->
    [...]
</dependencyManagement>
```



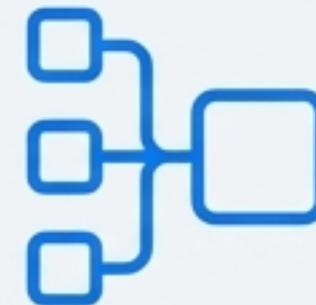
Lembre-se de remover esta dependência explícita ao atualizar para a próxima versão do CAP Java para voltar a receber a versão gerenciada pelo SDK.

Sua Estratégia para um Projeto Saudável e Seguro



Atualize Mensalmente

Adote a última versão `MINOR` da linha ativa para incorporar features e patches patches de segurança, minimizando o esforço de migração futuro.



Centralize com BOMs

Use os Bill of Materials (`cds-services-bom`, `sdk-modules-bom`, etc.) para garantir a consistência de versões e evitar conflitos.



Aja com Precisão

Saiba como sobreescrivar dependências de forma controlada para responder rapidamente a vulnerabilidades críticas sem desestabilizar o projeto.