

**Serviço Nacional de Aprendizagem Industrial**

**SENAI “Gaspar Ricardo Junior”**

**Curso**

**TÉCNICO EM DESENVOLVIMENTO**

**DE SISTEMAS**

# Métodos equals e hashCode em Java e o uso de Lombok para otimizar código em ambientes de desenvolvimento.

João Alexandre da Silva Pereira

Sorocaba

Novembro – 2024



**Serviço Nacional de Aprendizagem Industrial**

**SENAI “Gaspar Ricardo Junior”**

João Alexandre da Silva Pereira

# Métodos equals e hashCode em Java e o uso de Lombok para otimizar código em ambientes de desenvolvimento.

Profº Emerson Magalhães

Sorocaba

Março – 2024

**Sumário**

INTRODUÇÃO 3

1. COMO A UTILILIZAÇÃO DO IOT PODE TORNAR A PRODUÇÃO DAS INDÚSTRIAS MAIS EFICIENTE? 4

1.1. COMO SE COMUNICAM? 6

CONCLUSÃO 7

BIBLIOGRAFIA 8

* **INTRODUÇÃO**

**Contextualização dos métodos equals e hashCode:**

Introduza brevemente o que são esses métodos e onde eles se aplicam. Esclareça que equals determina a igualdade lógica entre objetos, enquanto hashCode fornece um valor numérico que facilita o uso em tabelas de hash.

Importância para coleções e frameworks: Destaque a importância desses métodos em coleções baseadas em hashing, como HashSet e HashMap, e o papel fundamental no comportamento correto de frameworks como o Spring.

Introdução ao Lombok: Explique o que é o Lombok e como ele auxilia na eliminação de código repetitivo, focando em como ele facilita a implementação de equals e hashCode.

* **Fundamentos Teóricos**

**Contrato entre equals e hashCode: Descreva as regras principais:**

Se dois objetos são iguais (conforme equals), seus hashCode devem ser iguais.

Se hashCode é implementado, equals também deve ser, para garantir consistência.

Impacto nas Coleções: Explore o impacto prático dessas regras em coleções que utilizam hashing:

Explicação de como HashMap e HashSet utilizam hashCode para distribuir e buscar dados.

Problemas que surgem ao violar o contrato, como objetos duplicados e buscas ineficientes.

Relevância para Entidades em Java: Discuta o uso correto de equals e hashCode em entidades persistentes (como no Spring Data), abordando aspectos como caching e operações de persistência.

**3. Utilização Prática em Coleções Java e no Spring**

Exemplos Práticos em Coleções: Forneça exemplos de uso com HashSet e HashMap:

Mostre como a implementação de equals e hashCode pode influenciar a inserção e busca de objetos nessas coleções.

Inclua exemplos de quando o contrato não é respeitado e como isso causa problemas.

Exemplo no Spring: Demonstre uma entidade do Spring, como uma classe de modelo usada no JPA, onde equals e hashCode são relevantes para garantir a identidade dos objetos em um contexto de cache ou quando recuperados do banco de dados.

* **4. Lombok: Simplificação do Código**

Introdução ao Lombok: Explique o que é a biblioteca e as anotações principais para geração automática de métodos.

Análise das Anotações @EqualsAndHashCode e @Data:

@EqualsAndHashCode: Gera automaticamente os métodos equals e hashCode, possibilitando a customização de quais campos devem ser incluídos.

@Data: Gera automaticamente equals, hashCode, toString, getters, setters e outros, sendo útil para classes simples.

Exemplo Prático com Lombok vs Implementação Manual: Compare uma implementação manual de equals e hashCode com o uso das anotações Lombok, destacando a economia de código e simplificação.

Vantagens e Desvantagens do Lombok:

Vantagens: Menos código boilerplate, maior legibilidade e manutenção simplificada.

Desvantagens: Dependência de uma biblioteca externa, o que pode afetar a depuração e a introspecção de código.Processo de Criação de Views no SQL

As Views são uma forma de armazenar uma consulta SQL em um objeto de banco de dados, permitindo fácil reutilização e simplificação do código. Elas ajudam a criar "visões" específicas dos dados, sem a necessidade de duplicar os dados no banco.

**5 - Discussão sobre Boas Práticas**

Lombok em Ambientes de Produção: Discuta a viabilidade de usar Lombok em ambientes de produção, incluindo pontos sobre a manutenção do código a longo prazo e o impacto sobre a clareza para novos desenvolvedores.

* **CONCLUSÃO**

O uso correto dos métodos equals, hashCode e de ferramentas como o Lombok tem um impacto significativo no desenvolvimento de aplicações Java, oferecendo uma série de benefícios, mas também apresentando desafios que devem ser cuidadosamente gerenciados.

Primeiramente, o contrato entre equals e hashCode é essencial para garantir o comportamento correto de objetos em coleções baseadas em hash, como HashMap ou HashSet. A implementação adequada desses métodos assegura que objetos sejam comparados de forma lógica e eficiente, o que é fundamental para a integridade das operações de busca, inserção e remoção nessas coleções. O uso consistente do contrato equals/hashCode também evita erros sutis, como a duplicação de objetos ou a perda de dados em estruturas de dados, melhorando a confiabilidade da aplicação.

Entretanto, o desafio de implementar esses métodos de maneira eficiente pode ser complexo, principalmente em classes com muitos atributos ou com hierarquias de objetos. O uso de bibliotecas como o Lombok facilita esse processo, oferecendo anotações como @EqualsAndHashCode, que geram automaticamente essas implementações, reduzindo o boilerplate e o risco de erros humanos. Além disso, Lombok contribui para a legibilidade e manutenibilidade do código, ao permitir que os desenvolvedores foquem nas funcionalidades essenciais sem se preocupar com a repetição de código.

Outro benefício do Lombok é a simplificação do código, que pode gerar automaticamente getters, setters, toString, equals e hashCode, entre outros, permitindo que os desenvolvedores mantenham o foco nas lógicas de negócios e não em detalhes de implementação. Isso resulta em um código mais enxuto, menos propenso a erros e mais fácil de manter a longo prazo. No entanto, é importante lembrar que o uso do Lombok, embora traga benefícios claros em termos de produtividade e simplificação, também requer cautela. A dependência de anotações pode obscurecer o comportamento de classes para desenvolvedores que não estão familiarizados com o Lombok, e pode exigir cuidados na escolha de quais métodos devem ser gerados automaticamente.

Em resumo, a implementação adequada de equals e hashCode, aliada ao uso de ferramentas como o Lombok, é fundamental para a criação de aplicações Java modernas e escaláveis. Esses conceitos garantem que o código seja eficiente, fácil de manter e livre de erros comuns que podem surgir em coleções baseadas em hash. Ao seguir o contrato equals/hashCode e utilizar recursos como o Lombok, os desenvolvedores são capazes de construir sistemas mais robustos, com menos redundância e maior clareza, promovendo a continuidade e o crescimento saudável das aplicações ao longo do tempo.

**BIBLIOGRAFIA**

[**https://sae.unb.br/cae/conteudo/unbfga/lbd/banco2\_visoes.html#:~:text=Em%20banco%20de%20dados%20relacionais,se%20fossem%20uma%20tabela%20real**](https://sae.unb.br/cae/conteudo/unbfga/lbd/banco2_visoes.html#:~:text=Em%20banco%20de%20dados%20relacionais,se%20fossem%20uma%20tabela%20real)

[**https://pt.stackoverflow.com/questions/35413/o-que-s%C3%A3o-views-em-sql-quais-vantagens-e-desvantagens-em-utilizar#:~:text=View%20%C3%A9%20um%20resultado%20originado,dados%20e%20n%C3%A3o%20cont%C3%A9m%20dados**](https://pt.stackoverflow.com/questions/35413/o-que-s%C3%A3o-views-em-sql-quais-vantagens-e-desvantagens-em-utilizar#:~:text=View%20%C3%A9%20um%20resultado%20originado,dados%20e%20n%C3%A3o%20cont%C3%A9m%20dados)