**RCA: Falha na Auditoria de Criação de Usuários no Sistema Fiberguardian**

**1. Resumo do Problema**

**Data do Incidente**: 14 de julho de 2025  
**Sistema**: Fiberguardian

**Componente Afetado**: Auditoria de entidades (Spring Data JPA)  
**Descrição**: Durante o cadastro de novos usuários no endpoint /usuarios, a coluna criado\_por (anotada com @CreatedBy) na entidade Usuario estava sendo gravada como null no banco de dados. Além disso, a classe AuditorAwareImpl não estava sendo invocada, conforme indicado pela ausência de logs esperados (e.g., #################### 🚀 AuditorAwareImpl carregado.).

**2. Impacto**

* **Funcionalidade Afetada**: A rastreabilidade de quem criou um usuário não estava sendo registrada, violando os requisitos de auditoria e rastreabilidade (OWASP A10: Insufficient Logging & Monitoring).
* **Usuários Impactados**: Administradores que dependem da coluna criado\_por para rastrear ações no sistema.
* **Severidade**: Moderada, pois a funcionalidade principal de cadastro funcionava, mas a auditoria estava incompleta.

**3. Cronologia**

* \*\*16:09:55, 14最后一

System: 14/07/2025\*\*: Um administrador autenticado realizou login (ana.souza@fiberguardian.com) e tentou criar um novo usuário via endpoint /usuarios. O log mostrou que a query de inserção (insert into usuario) foi executada, mas a coluna criado\_por estava null.

* **Investigação**: Verificação do código revelou que a classe AuditorAwareImpl não estava sendo chamada, e a entidade Usuario não possuía a anotação @EntityListeners(AuditingEntityListener.class).

**4. Causa Raiz**

* **Causa Principal**: A entidade Usuario não estava anotada com @EntityListeners(AuditingEntityListener.class), necessária para que o Spring Data JPA processe as anotações de auditoria (@CreatedBy, @CreationTimestamp, @LastModifiedBy, @UpdateTimestamp).
* **Detalhes**: Sem essa anotação, o AuditingEntityListener não foi acionado, impedindo que o AuditorAwareImpl fornecesse o ID do usuário autenticado para a coluna criado\_por. O log confirmou que o SecurityContextHolder estava corretamente populado com um UsuarioAutenticado, mas a ausência do listener bloqueou a auditoria.

**5. Ação Corretiva**

* **Solução Aplicada**:
  + Adicionada a anotação @EntityListeners(AuditingEntityListener.class) à entidade Usuario:
  + @Entity
  + @EntityListeners(AuditingEntityListener.class)
  + public class Usuario {
  + // ... campos e anotações
  + }
  + Após a adição, a auditoria passou a funcionar corretamente, preenchendo a coluna criado\_por com o ID do usuário autenticado (e.g., 1 para ana.souza@fiberguardian.com).
* **Comportamento Observado**: As colunas data\_alteracao (@UpdateTimestamp) e alterado\_por (@LastModifiedBy) também foram preenchidas na criação com os mesmos valores de data\_cadastro e criado\_por, respectivamente, devido ao comportamento padrão do Spring Data JPA.

**6. Decisão Sobre Comportamento Padrão**

* **Observação**: O preenchimento das colunas data\_alteracao e alterado\_por na criação foi identificado como comportamento padrão do Spring Data JPA, que considera a criação uma modificação inicial.
* **Decisão**: Optou-se por manter o comportamento padrão, pois é simples, consistente com outras entidades e não compromete a rastreabilidade (alinhado com OWASP).
* **Alternativa Considerada**: Implementar um CustomAuditingEntityListener para garantir que data\_alteracao e alterado\_por sejam null na criação, mas foi descartada por ser desnecessária no contexto do sistema.

**7. Justificativas**

* **Segurança (OWASP)**: A solução garante rastreabilidade completa das ações de criação e modificação de usuários, atendendo aos requisitos de auditoria.
* **KISS**: A adição de @EntityListeners é uma solução padrão e simples, evitando complexidade desnecessária.
* **SOLID**: A responsabilidade de auditoria foi mantida no AuditingEntityListener, enquanto a entidade Usuario foca na lógica de domínio.
* **Manutenibilidade**: A documentação clara e o uso de logs (adicionados em JpaConfig e AuditorAwareImpl) facilitam a depuração futura.

**8. Recomendações para Prevenção**

* **Verificação de Configuração de Auditoria**:
  + Sempre confirmar que entidades auditáveis possuem @EntityListeners(AuditingEntityListener.class) ao usar anotações como @CreatedBy e @LastModifiedBy.
  + Adicionar logs em classes de configuração (e.g., JpaConfig, AuditorAwareImpl) para validar a inicialização:
  + logger.info("JpaConfig inicializado com auditoria habilitada.");
  + logger.info("AuditorAwareImpl inicializado.");
* **Testes de Integração**:
  + Implementar testes para validar o comportamento da auditoria, como:
  + @Test
  + public void testCriaUsuarioComAuditoria() {
  + SecurityContextHolder.getContext().setAuthentication(
  + new UsernamePasswordAuthenticationToken(
  + new UsuarioAutenticado(1L, "ana.souza@fiberguardian.com"),
  + null,
  + Collections.singletonList(new SimpleGrantedAuthority("ROLE\_ADMIN"))
  + )
  + );
  + Usuario usuario = new Usuario();
  + usuario.setNome("Novo Usuário");
  + usuario.setEmail("novo@fiberguardian.com");
  + usuario = usuarioService.criarUsuario(usuario);
  + assertNotNull(usuario.getCriadoPor());
  + assertEquals(1L, usuario.getCriadoPor());
  + }
* **Revisão de Dependências**:
  + Garantir que o pom.xml inclua as dependências necessárias:
  + <dependency>
  + <groupId>org.springframework.boot</groupId>
  + <artifactId>spring-boot-starter-data-jpa</artifactId>
  + </dependency>
  + <dependency>
  + <groupId>org.springframework.boot</groupId>
  + <artifactId>spring-boot-starter-security</artifactId>
  + </dependency>
* **Documentação**:
  + Manter documentação detalhada sobre o comportamento das colunas de auditoria, incluindo a observação de que @UpdateTimestamp e @LastModifiedBy são preenchidos na criação por padrão.

**9. Conclusão**

O problema foi resolvido com a adição de @EntityListeners(AuditingEntityListener.class) à entidade Usuario, permitindo que o Spring Data JPA processe as anotações de auditoria corretamente. O comportamento padrão de preenchimento de data\_alteracao e alterado\_por na criação foi mantido por ser simples e alinhado com os requisitos do sistema. Recomendações de prevenção, incluindo logs, testes e verificação de configurações, foram estabelecidas para evitar problemas semelhantes no futuro.