

Testes para “Cadastro de Usuário”

1. Descrição da Funcionalidade

- Tela para inserir, excluir e atualizar usuários.
- Campos: nome (obrigatório), email (obrigatório e único), endereço, telefone.
- Apenas administradores podem excluir usuários.

2. Tipos de Testes

2.1 Testes Unitários

- Focam em validar regras de negócio isoladas.
- Exemplos: validação de campos obrigatórios, formato de email, unicidade.

2.2 Testes de Integração

- Verificam interação com camada de persistência e API.
- Use banco em memória (H2) ou testes de contrato REST.

2.3 Testes End-to-End (E2E)

- Simulam fluxo completo via UI (Selenium, Cypress).
- Cadastram, editam e excluem usuários, validando respostas da UI.

3. Cenários de Teste

- CT01: Cadastro válido → status 201, usuário criado no banco.
- CT02: Nome vazio → status 400, erro 'Nome é obrigatório'.
- CT03: Email inválido → status 400, erro 'Email inválido'.
- CT04: Email duplicado → status 409, erro 'Email já cadastrado'.
- CT05: Atualização válida → status 200, dados atualizados.
- CT06: Exclusão por ADMIN → status 200, usuário removido.
- CT07: Exclusão por não-ADMIN → status 403, operação proibida.
- CT08: Entrada com campos muito longos → status 400, erros de tamanho.

4. Exemplo de Teste Unitário (JUnit + Spring Boot)

```
public class UserServiceTest {
    @SpringBootTest
    public class UserServiceTest {
        @Autowired
        private UserService userService;

        @Test
        void givenValidUser_whenCreate_thenReturnUser() {
            UserDto dto = new UserDto("Alice", "alice@example.com", "Rua A", "12345");
            User u = userService.create(dto);
            assertNotNull(u.getId());
            assertEquals("Alice", u.getName());
        }
    }
}
```

5. Boas Práticas

- Use mocks para serviços externos em testes unitários.
- Limpe o banco de dados entre testes de integração.
- Automatize E2E em pipeline CI/CD.

Conclusão: Uma cobertura adequada de testes garante confiabilidade e qualidade no cadastro de usuários.