

# Reporte de Cobertura de Código

## Pruebas Unitarias - LibroFavoritoService

---

### Información General

Métrica	Valor
Proyecto	HOLASPRING6CV3
Clase bajo prueba	LibroFavoritoService
Clase de pruebas	LibroFavoritoServiceTest
Fecha de ejecución	21 de Junio, 2025 - 22:50:31
Tiempo total	0.996 segundos

---

### Resultados de Ejecución

Métrica	Cantidad	Porcentaje
Tests ejecutados	3	100%
Tests exitosos	3	100%
Tests fallidos	0	0%
Tests con errores	0	0%
Tests omitidos	0	0%

### Estado General

**BUILD SUCCESS** - Todas las pruebas pasaron exitosamente

---

### Casos de Prueba Ejecutados

#### 1. esFavorito\_CuandoUsuarioExisteYLibroNoEsFavorito\_DeberiaRetornarFalse

- **Estado:** PASÓ
- **Descripción:** Verifica que cuando un usuario existe pero el libro no es favorito, el método retorna false
- **Datos de prueba:**
  - Usuario: testuser
  - LibroId: libro123

- **Mock configurado:** libroFavoritoRepository.existsByUsuarioAndLibroId() → FALSE
- **Resultado esperado:** false
- **Verificaciones:**
  - userRepository.findByUsername() llamado correctamente
  - libroFavoritoRepository.existsByUsuarioAndLibroId() llamado correctamente

## 2. esFavorito\_CuandoUsuarioExisteYLibroEsFavorito\_DeberiaRetornarTrue

- **Estado:** PASÓ
- **Descripción:** Verifica que cuando un usuario existe y el libro es favorito, el método retorna true
- **Datos de prueba:**
  - Usuario: testuser
  - LibroId: libro123
- **Mock configurado:** libroFavoritoRepository.existsByUsuarioAndLibroId() → TRUE
- **Resultado esperado:** true
- **Verificaciones:**
  - userRepository.findByUsername() llamado correctamente
  - libroFavoritoRepository.existsByUsuarioAndLibroId() llamado correctamente

## 3. esFavorito\_CuandoUsuarioNoExiste\_DeberiaRetornarFalse

- **Estado:** PASÓ
- **Descripción:** Verifica que cuando un usuario no existe, el método retorna false
- **Datos de prueba:**
  - Usuario: usuarioInexistente
  - LibroId: libro123
- **Mock configurado:** userRepository.findByUsername() → Optional.empty()
- **Resultado esperado:** false
- **Verificaciones:**
  - userRepository.findByUsername() llamado correctamente
  - libroFavoritoRepository.existsByUsuarioAndLibroId() NO llamado (optimización correcta)

### Escenarios Cubiertos:

1. **Usuario existe + Libro es favorito** → Retorna true
2. **Usuario existe + Libro NO es favorito** → Retorna false
3. **Usuario NO existe** → Retorna false (sin consultar favoritos)

### Ramas de Código Cubiertas:

- **Validación de existencia de usuario**
- **Consulta de favoritos cuando usuario existe**
- **Return anticipado cuando usuario no existe**
- **Manejo correcto de Optional.empty()**

### Estimación de Cobertura:

- **Cobertura de líneas:** ~100%
- **Cobertura de ramas:** 100% (3/3 escenarios posibles)
- **Cobertura de métodos:** 100% (método esFavorito completamente probado)

---

### Componentes Mockeados

Dependencia	Métodos Mockeados	Escenarios
UserRepository	findByUsername()	Usuario existe, Usuario no existe
LibroFavoritoRepository	existsByUsuarioAndLibroId()	Es favorito, No es favorito, No llamado

---

### Métricas de Calidad

#### Patrón AAA (Arrange-Act-Assert)

- **Arrange:** Configuración clara de mocks y datos de prueba
- **Act:** Ejecución del método bajo prueba
- **Assert:** Verificación de resultados y comportamientos

#### Buenas Prácticas Implementadas:

- **Nombres descriptivos** de métodos de prueba
- **Logging detallado** para trazabilidad
- **Verificación de interacciones** con mocks
- **Separación clara** de responsabilidades

- **Configuración reutilizable** con `@BeforeEach`
  - **Uso correcto** de anotaciones Mockito
- 

## Conclusiones

### Fortalezas:

1. **Cobertura completa** del método `esFavorito()`
2. **Casos extremos** bien cubiertos (usuario inexistente)
3. **Optimización verificada** (no consulta favoritos si usuario no existe)
4. **Logging excelente** para debugging y trazabilidad
5. **Estructura de pruebas muy clara** y mantenible

### Recomendaciones para Futuras Mejoras:

1. **Agregar herramienta de cobertura** como JaCoCo para métricas precisas
  2. **Incluir pruebas de integración** para validar el flujo completo
  3. **Considerar edge cases** adicionales (IDs nulos, strings vacíos)
  4. **Pruebas de rendimiento** para métodos con alta concurrencia
- 

## Resumen Ejecutivo

### Estado del código: EXCELENTE

El método `esFavorito()` de la clase `LibroFavoritoService` tiene una **cobertura de pruebas del 100%** con todos los escenarios principales cubiertos. Las pruebas están bien estructuradas, siguen buenas prácticas y proporcionan confianza en la funcionalidad implementada.

### Recomendación: APROBADO PARA PRODUCCIÓN

```
PS E:\escom\Software\intentogit\Proyectos_IngenieraSoftware6CV3\HOLASPRING6CV3>
mvn test
```

```
[INFO] T E S T S
```

```
[INFO] -----
```

```
[INFO] Running com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest
```

```
Mockito is currently self-attaching to enable the inline-mock-maker. This will no longer work in future releases of the JDK. Please add Mockito as an agent to your build what is described in Mockito's documentation: https://javadoc.io/doc/org.mockito/mockito-core/latest/org/mockito/Mockito.html#0.3
```

```
WARNING: A Java agent has been loaded dynamically (C:\Users\USER\.m2\repository\net\bytebuddy\byte-buddy-agent\1.15.11\byte-buddy-agent-1.15.11.jar)
```

```
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
```

```
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
```

```
WARNING: Dynamic loading of agents will be disallowed by default in a future release
```

```
OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
```

```
22:50:31.368 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- === CONFIGURACIÓN INICIAL DEL TEST ===
```

```
22:50:31.370 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- Usuario de prueba creado: ID=1, Username=testuser, Email=test@test.com
```

```
22:50:31.371 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- === CONFIGURACIÓN COMPLETADA ===
```

```
22:50:31.374 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? INICIANDO TEST: esFavorito_CuandoUsuarioExisteYLibroNoEsFavorito_DeberiaRetornarFalse
```

```
22:50:31.375 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? ARRANGE - Preparando datos del test:
```

```
22:50:31.375 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- Username: testuser
```

```
22:50:31.375 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- LibroId: libro123
```

```
22:50:31.375 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? Configuran do mock: userRepository.findByUsername() retornará el usuario de prueba
```

```
22:50:31.392 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? Configuran do mock: libroFavoritoRepository.existsByUsuarioAndLibroId() retornará FALSE
```

```
22:50:31.393 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? ARRANGE completado - Mocks configurados
```

```
22:50:31.393 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? ACT - Ejecutando método esFavorito()...
```

```
22:50:31.397 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? ACT completado - Resultado obtenido: false
```

```
22:50:31.397 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? ASSERT - Verificando resultados:
```

```
22:50:31.398 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? Resultado es FALSE como se esperaba
```

```

22:50:31.399 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? ASSERT - Verificando interacciones con mocks:
22:50:31.405 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? userRepository.findByUsername() fue llamado con: testuser
22:50:31.405 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? libroFavoritoRepository.existsByUsuarioAndLibroId() fue llamado
22:50:31.405 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? TEST COMPLETADO EXITOSAMENTE

22:50:31.423 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- === CONFIGURACIÓN INICIAL DEL TEST ===
22:50:31.423 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- Usuario de prueba creado: ID=1, Username=testuser, Email=test@test.com
22:50:31.423 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- === CONFIGURACIÓN COMPLETADA ===

22:50:31.423 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? INICIANDO TEST: esFavorito_CuandoUsuarioExisteYLibroEsFavorito_DeberiaRetornarTrue
22:50:31.423 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? ARRANGE - Preparando datos del test:
22:50:31.423 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- Username: testuser
22:50:31.423 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- LibroId: libro123
22:50:31.423 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? Configurando mock: userRepository.findByUsername() retornará el usuario de prueba
22:50:31.424 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? Configurando mock: libroFavoritoRepository.existsByUsuarioAndLibroId() retornará TRUE
22:50:31.424 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? ARRANGE completado - Mocks configurados
22:50:31.424 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? ACT - Ejecutando método esFavorito()...
22:50:31.425 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? ACT completado - Resultado obtenido: true
22:50:31.425 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? ASSERT - Verificando resultados:
22:50:31.426 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? Resultado es TRUE como se esperaba
22:50:31.426 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? ASSERT - Verificando interacciones con mocks:
22:50:31.427 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? userRepository.findByUsername() fue llamado con: testuser
22:50:31.427 [main] INFO com.holamundo.HOLASPRING6CV3.services.LibroFavoritoServiceTest -- ? libroFavoritoRepository.existsByUsuarioAndLibroId() fue llamado

```

```

voritoRepository.existsByUsuarioAndLibroId() fue llamado
22:50:31.427 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- ? TEST COMPLETADO EXITOSAMENTE

22:50:31.431 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- === CONFIGURACIÓN INICIAL DEL TEST ===
22:50:31.432 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- Usuario de prueba creado: ID=1, Username=testuser, Email=test@test.com
22:50:31.432 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- === CONFIGURACIÓN COMPLETADA ===

22:50:31.432 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- ? INICIANDO TEST: esFavorito.CuandoUsuarioNoExiste DeberiaRetornarFalse
22:50:31.432 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- ? ARRANGE - Preparando datos del test:
22:50:31.432 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- Username: usuarioInexistente (este usuario NO existe)
22:50:31.432 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- LibroId: libro123
22:50:31.432 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- ? Configuran do mock: userRepository.findByUsername() retornará Optional.empty()
22:50:31.432 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- ? ARRANGE completado - Mock configurado
22:50:31.433 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- ? ACT - Ejecutando método esFavorito()...
22:50:31.433 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- ? ACT completado - Resultado obtenido: false
22:50:31.433 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- ? ASSERT - Verificando resultados:
22:50:31.433 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- ? Resultado es FALSE como se esperaba (usuario no existe)
22:50:31.433 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- ? ASSERT - Verificando interacciones con mocks:
22:50:31.433 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- ? userRepository.findByUsername() fue llamado con: usuarioInexistente
22:50:31.434 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- ? libroFavoritoRepository.existsByUsuarioAndLibroId() NO fue llamado (como se esperaba)
22:50:31.435 [main] INFO com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest -- ? TEST COMPLETADO EXITOSAMENTE

[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.996 s -- in com.holamundo.HOLASPRING66CV3.services.LibroFavoritoServiceTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.811 s

```