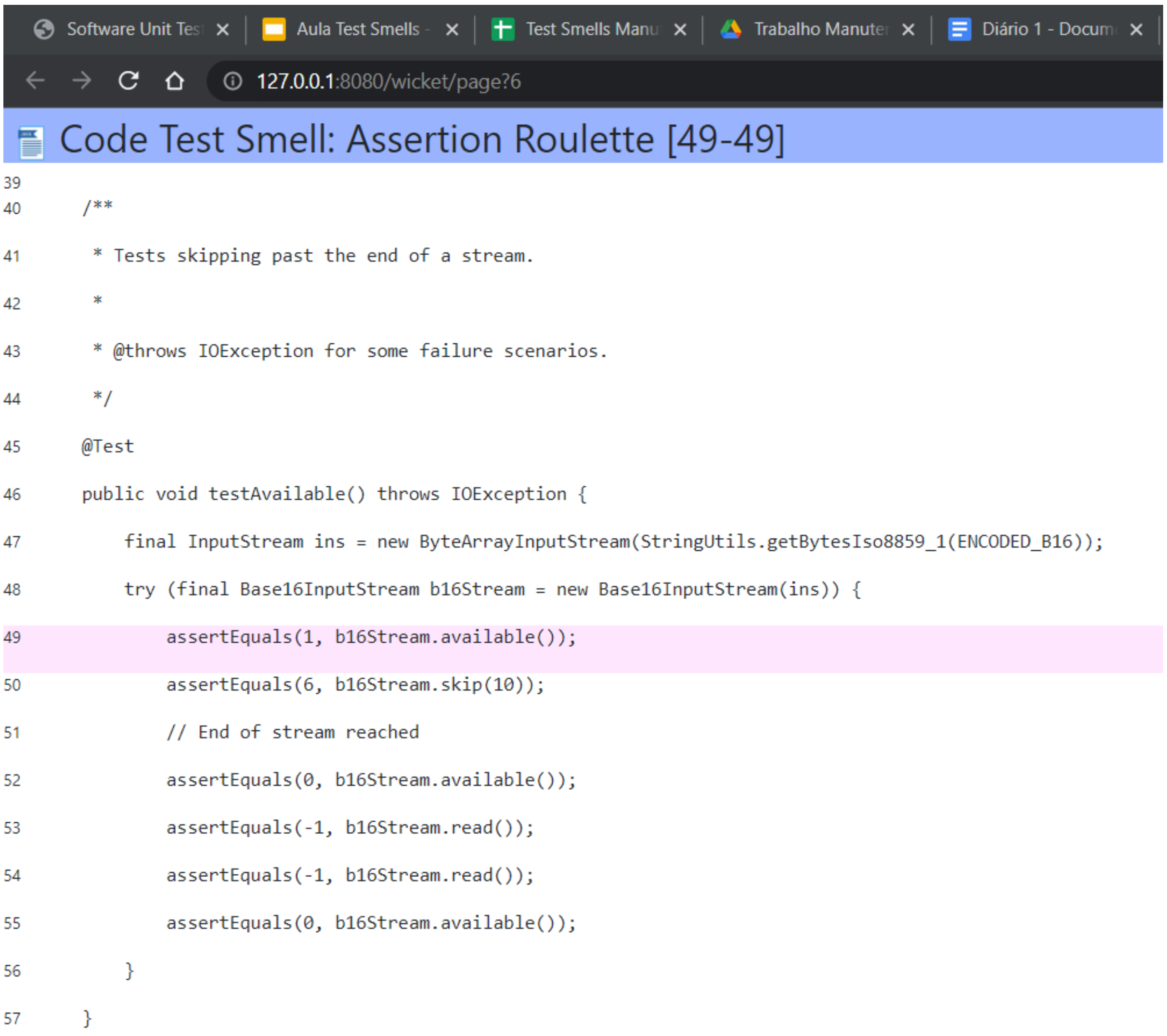
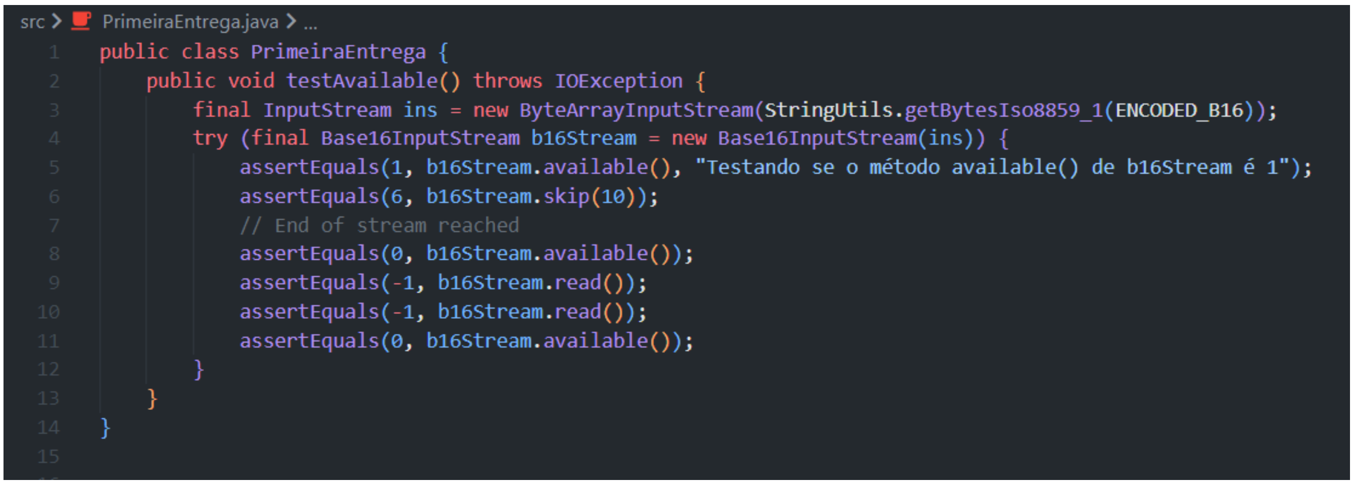
Assertion Roulette - 1 (linha 49-49)



Code Test Smell: Assertion Roulette [49-49]

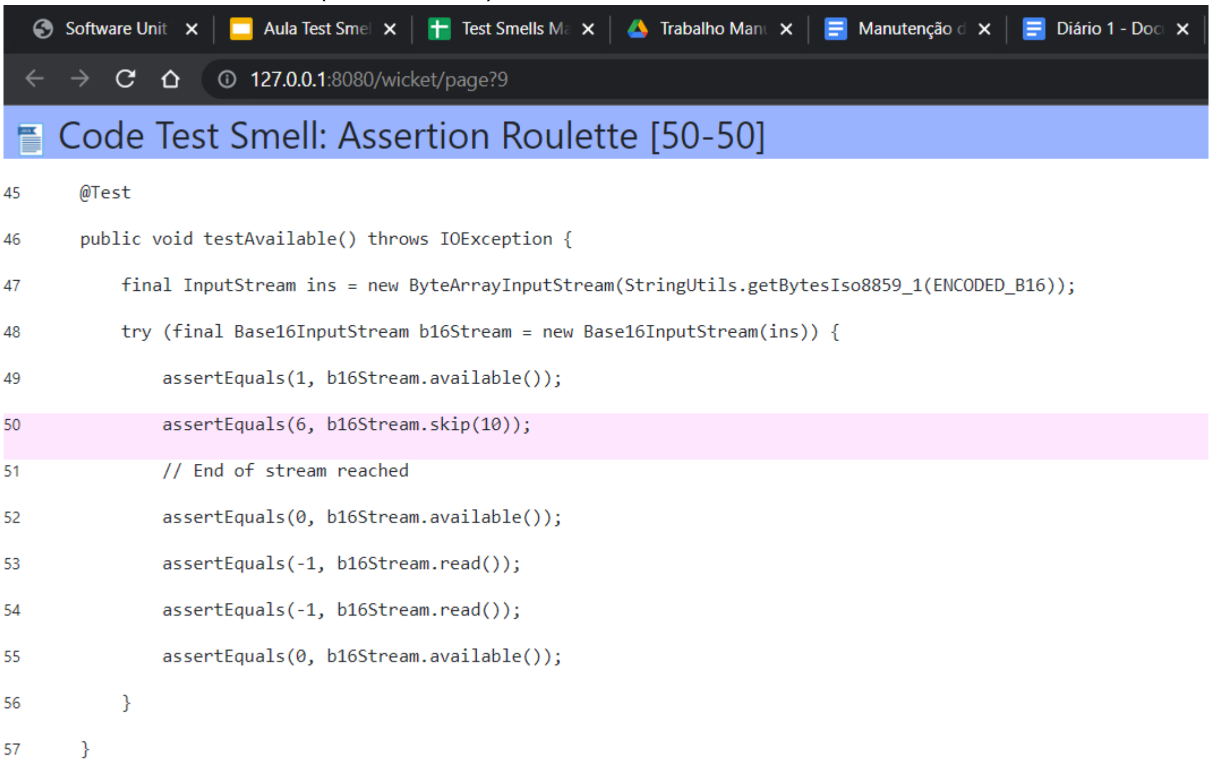
```
    /**
     * Tests skipping past the end of a stream.
     *
     * @throws IOException for some failure scenarios.
     */
    @Test
    public void testAvailable() throws IOException {
        final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
        try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
            assertEquals(1, b16Stream.available());
            assertEquals(6, b16Stream.skip(10));
            // End of stream reached
            assertEquals(0, b16Stream.available());
            assertEquals(-1, b16Stream.read());
            assertEquals(-1, b16Stream.read());
            assertEquals(0, b16Stream.available());
        }
    }
```

Teste refatorado



```
public class PrimeiraEntrega {
    public void testAvailable() throws IOException {
        final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
        try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
            assertEquals(1, b16Stream.available(), "Testando se o método available() de b16Stream é 1");
            assertEquals(6, b16Stream.skip(10));
            // End of stream reached
            assertEquals(0, b16Stream.available());
            assertEquals(-1, b16Stream.read());
            assertEquals(-1, b16Stream.read());
            assertEquals(0, b16Stream.available());
        }
    }
}
```

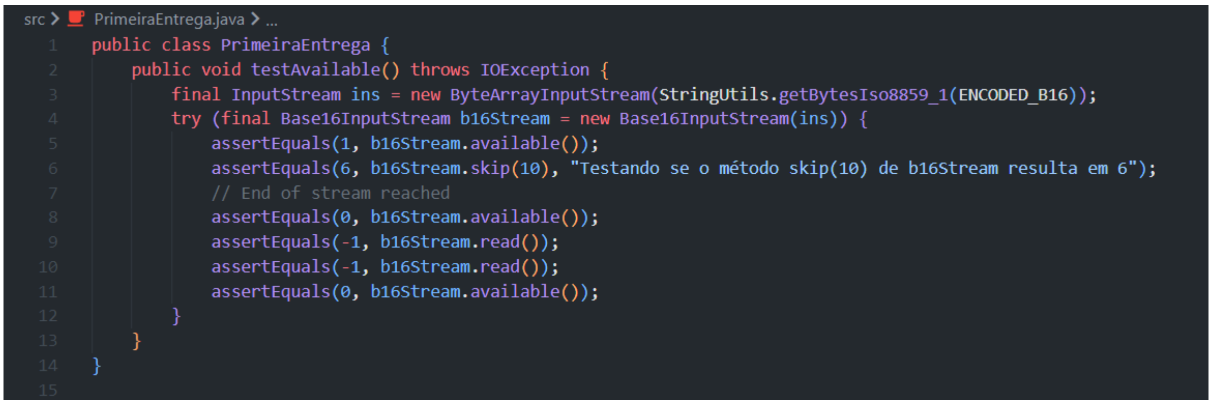
Assertion Roulette - 2 (linha 50-50)



Code Test Smell: Assertion Roulette [50-50]

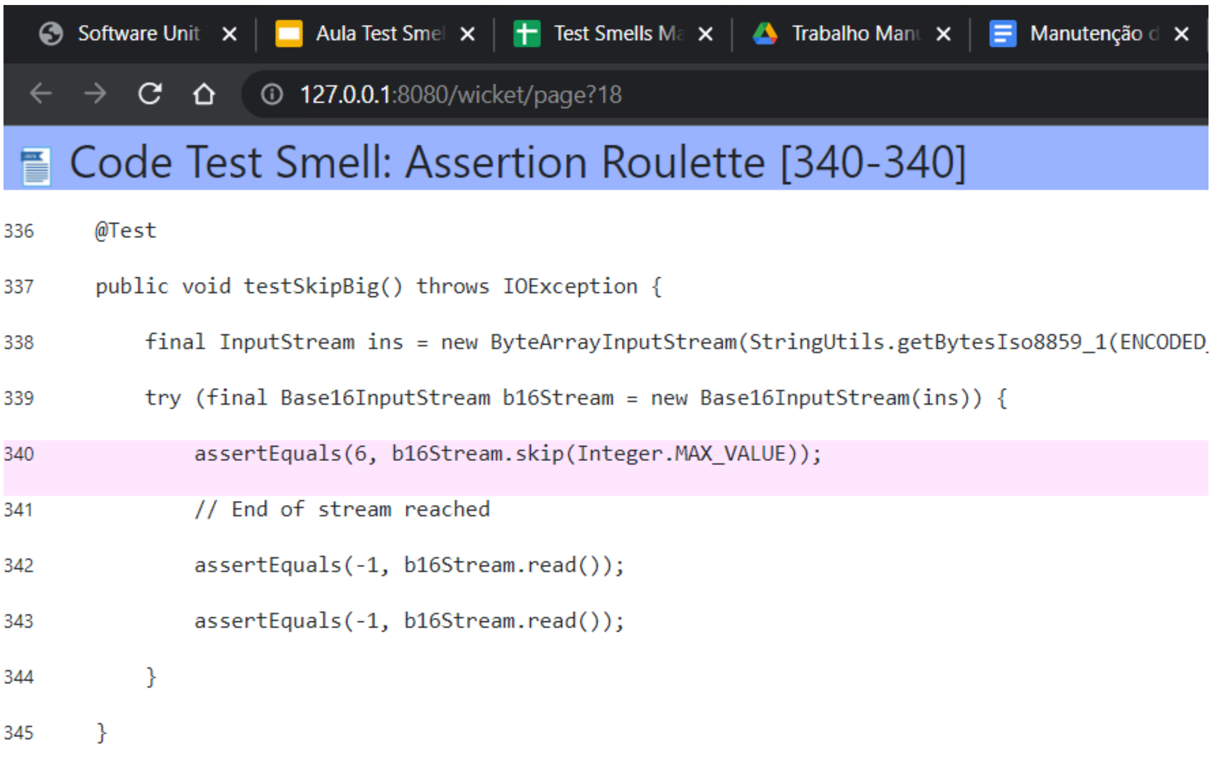
```
45    @Test
46    public void testAvailable() throws IOException {
47        final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
48        try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
49            assertEquals(1, b16Stream.available());
50            assertEquals(6, b16Stream.skip(10));
51            // End of stream reached
52            assertEquals(0, b16Stream.available());
53            assertEquals(-1, b16Stream.read());
54            assertEquals(-1, b16Stream.read());
55            assertEquals(0, b16Stream.available());
56        }
57    }
```

Teste refatorado:



```
public class PrimeiraEntrega {
    public void testAvailable() throws IOException {
        final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
        try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
            assertEquals(1, b16Stream.available());
            assertEquals(6, b16Stream.skip(10), "Testando se o método skip(10) de b16Stream resulta em 6");
            // End of stream reached
            assertEquals(0, b16Stream.available());
            assertEquals(-1, b16Stream.read());
            assertEquals(-1, b16Stream.read());
            assertEquals(0, b16Stream.available());
        }
    }
}
```

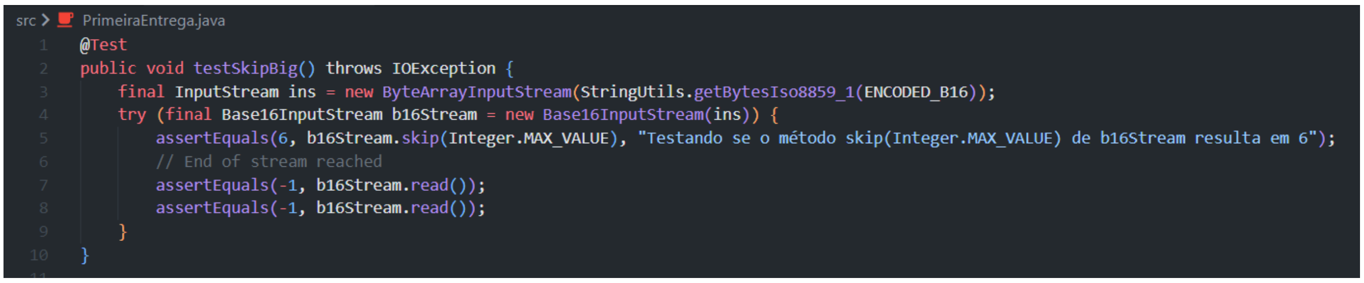
Assertion Roulette - 3 (linha 340-340)



Code Test Smell: Assertion Roulette [340-340]

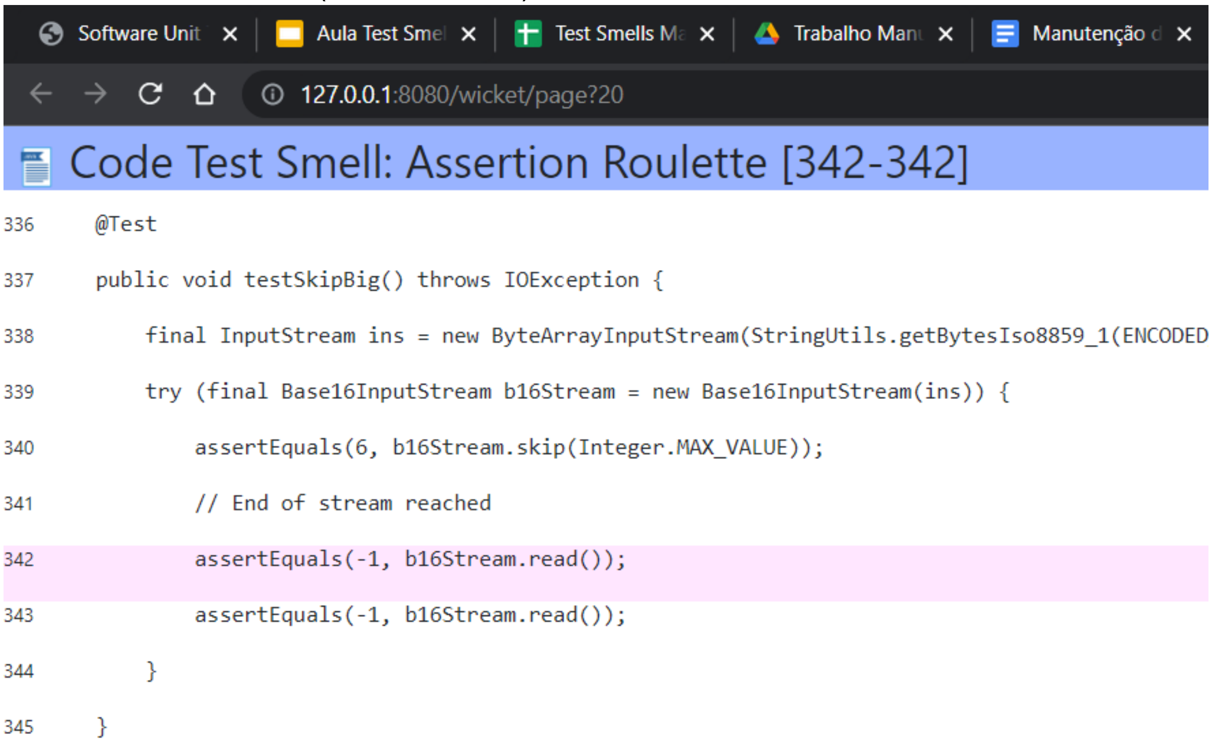
```
336    @Test
337    public void testSkipBig() throws IOException {
338        final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_
339        try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
340            assertEquals(6, b16Stream.skip(Integer.MAX_VALUE));
341            // End of stream reached
342            assertEquals(-1, b16Stream.read());
343            assertEquals(-1, b16Stream.read());
344        }
345    }
```

Teste refatorado



```
@Test
public void testSkipBig() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(6, b16Stream.skip(Integer.MAX_VALUE), "Testando se o método skip(Integer.MAX_VALUE) de b16Stream resulta em 6");
        // End of stream reached
        assertEquals(-1, b16Stream.read());
        assertEquals(-1, b16Stream.read());
    }
}
```

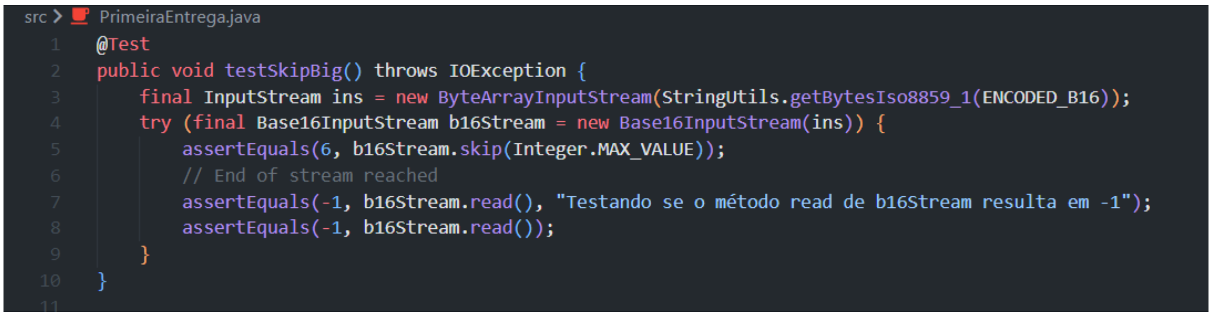
Assertion Roulette - 4 (linha 342-342)



Code Test Smell: Assertion Roulette [342-342]

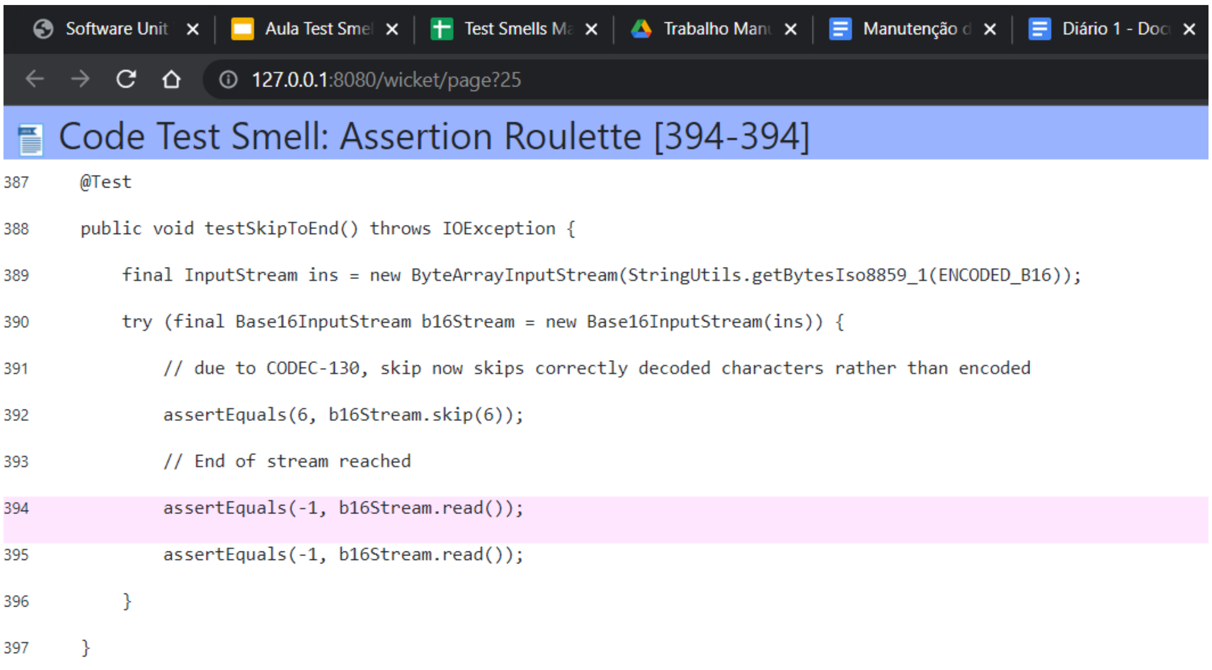
```
336     @Test
337     public void testSkipBig() throws IOException {
338         final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED
339         try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
340             assertEquals(6, b16Stream.skip(Integer.MAX_VALUE));
341             // End of stream reached
342             assertEquals(-1, b16Stream.read());
343             assertEquals(-1, b16Stream.read());
344         }
345     }
```

Teste refatorado



```
@Test
public void testSkipBig() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(6, b16Stream.skip(Integer.MAX_VALUE));
        // End of stream reached
        assertEquals(-1, b16Stream.read(), "Testando se o método read de b16Stream resulta em -1");
        assertEquals(-1, b16Stream.read());
    }
}
```

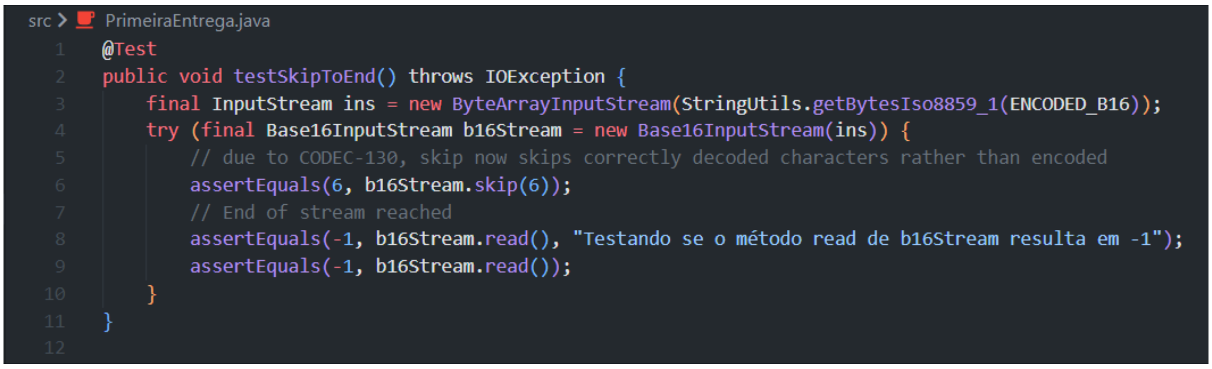
Assertion Roulette - 5 (linha 394-394)



Code Test Smell: Assertion Roulette [394-394]

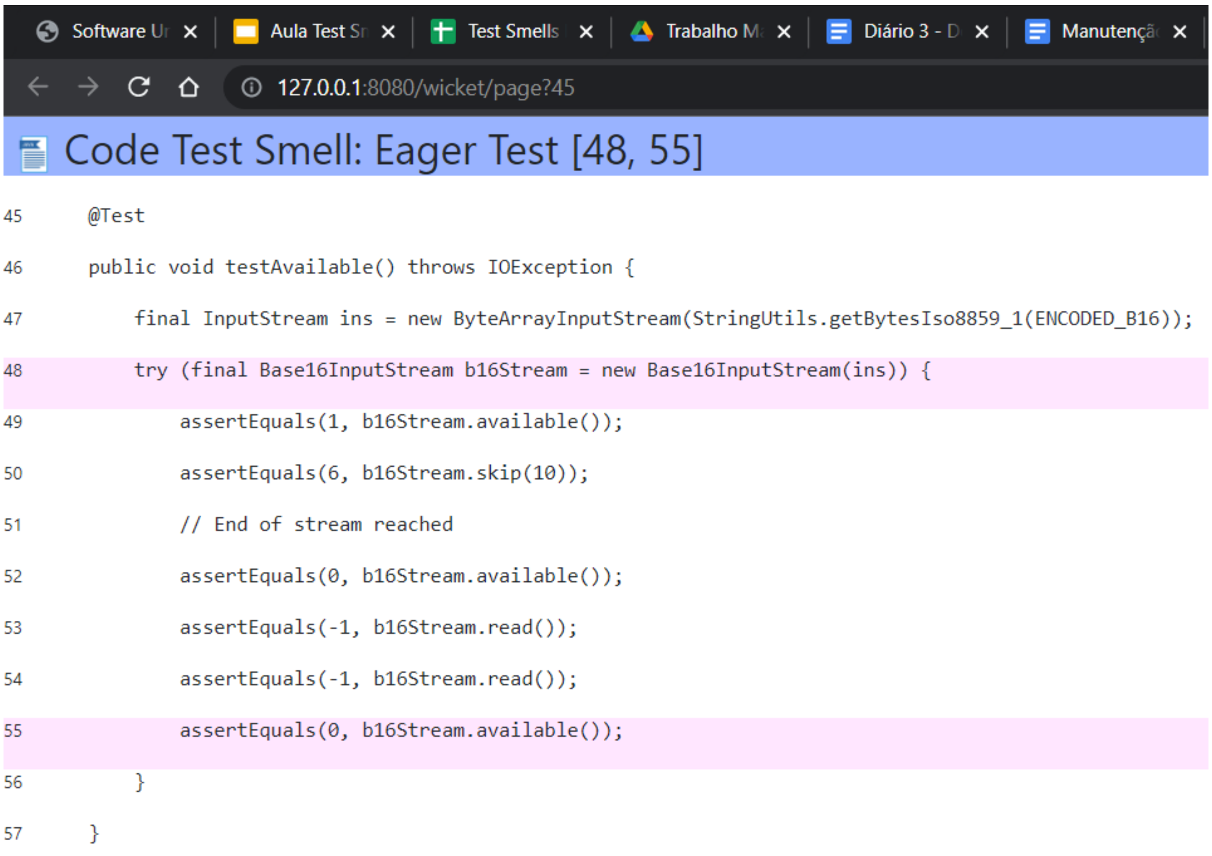
```
@Test
public void testSkipToEnd() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        // due to CODEC-130, skip now skips correctly decoded characters rather than encoded
        assertEquals(6, b16Stream.skip(6));
        // End of stream reached
        assertEquals(-1, b16Stream.read());
        assertEquals(-1, b16Stream.read());
    }
}
```

Teste refatorado



```
@Test
public void testSkipToEnd() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        // due to CODEC-130, skip now skips correctly decoded characters rather than encoded
        assertEquals(6, b16Stream.skip(6));
        // End of stream reached
        assertEquals(-1, b16Stream.read(), "Testando se o método read de b16Stream resulta em -1");
        assertEquals(-1, b16Stream.read());
    }
}
```

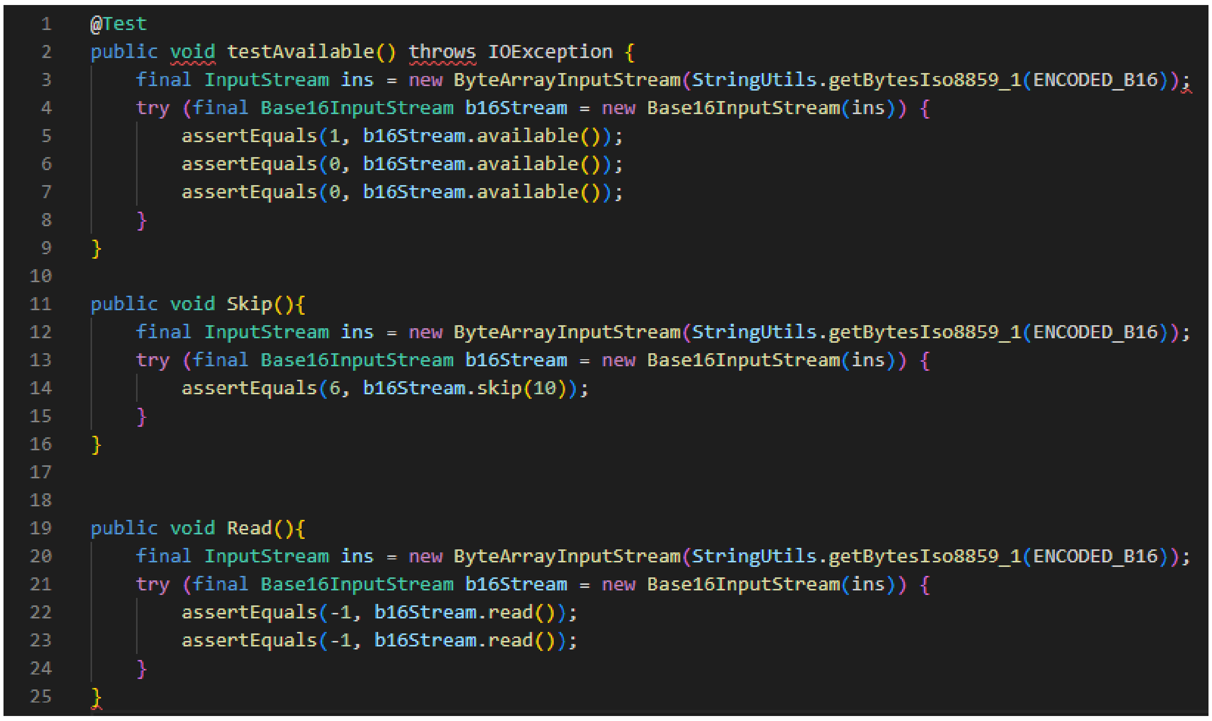
Eager Test - 1 (linha 48-55)



Code Test Smell: Eager Test [48, 55]

```
45    @Test
46    public void testAvailable() throws IOException {
47        final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
48        try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
49            assertEquals(1, b16Stream.available());
50            assertEquals(6, b16Stream.skip(10));
51            // End of stream reached
52            assertEquals(0, b16Stream.available());
53            assertEquals(-1, b16Stream.read());
54            assertEquals(-1, b16Stream.read());
55            assertEquals(0, b16Stream.available());
56        }
57    }
```

Teste refatorado

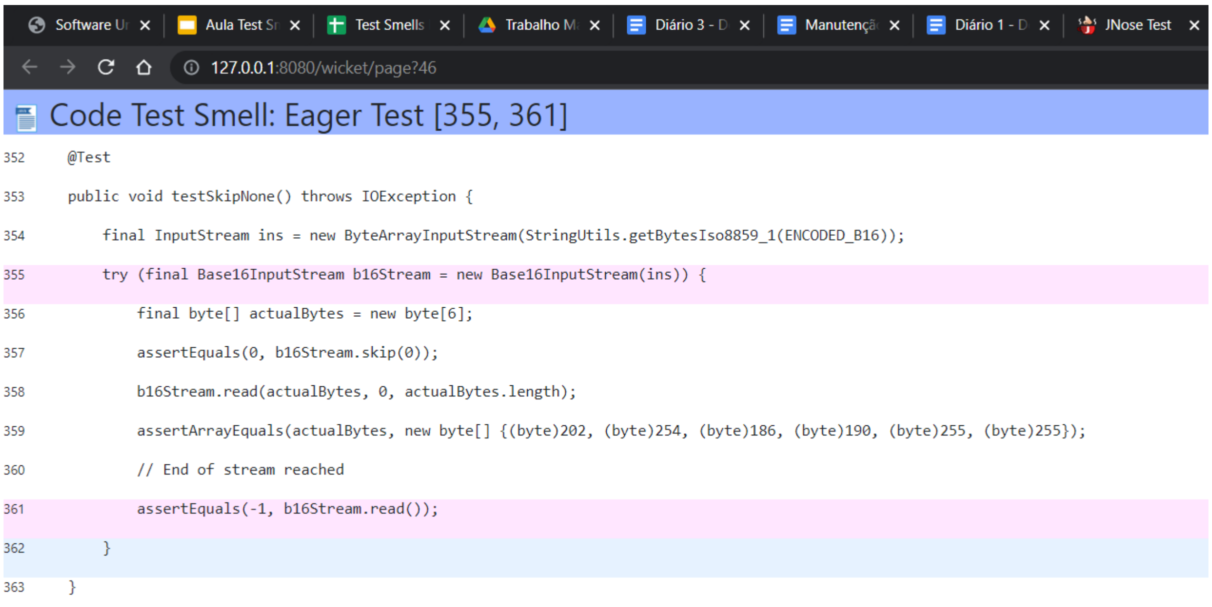


```
@Test
public void testAvailable() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(1, b16Stream.available());
        assertEquals(0, b16Stream.available());
        assertEquals(0, b16Stream.available());
    }
}

public void Skip(){
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(6, b16Stream.skip(10));
    }
}


public void Read(){
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(-1, b16Stream.read());
        assertEquals(-1, b16Stream.read());
    }
}
```

Eager Test - 2 (linha 355-361)

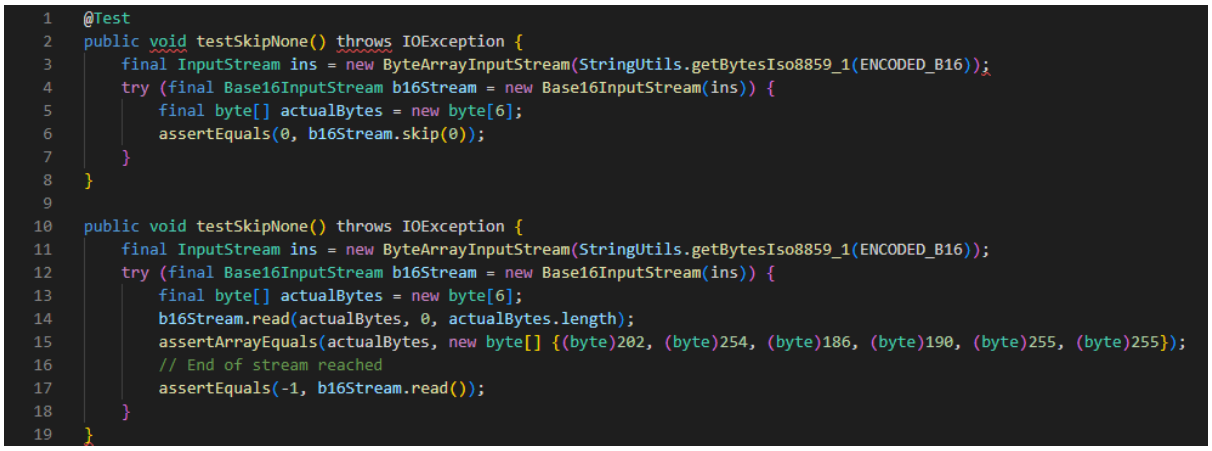


127.0.0.1:8080/wicket/page?46

Code Test Smell: Eager Test [355, 361]

```
352     @Test
353     public void testSkipNone() throws IOException {
354         final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
355         try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
356             final byte[] actualBytes = new byte[6];
357             assertEquals(0, b16Stream.skip(0));
358             b16Stream.read(actualBytes, 0, actualBytes.length);
359             assertArrayEquals(actualBytes, new byte[] {(byte)202, (byte)254, (byte)186, (byte)190, (byte)255, (byte)255});
360             // End of stream reached
361             assertEquals(-1, b16Stream.read());
362         }
363     }
```

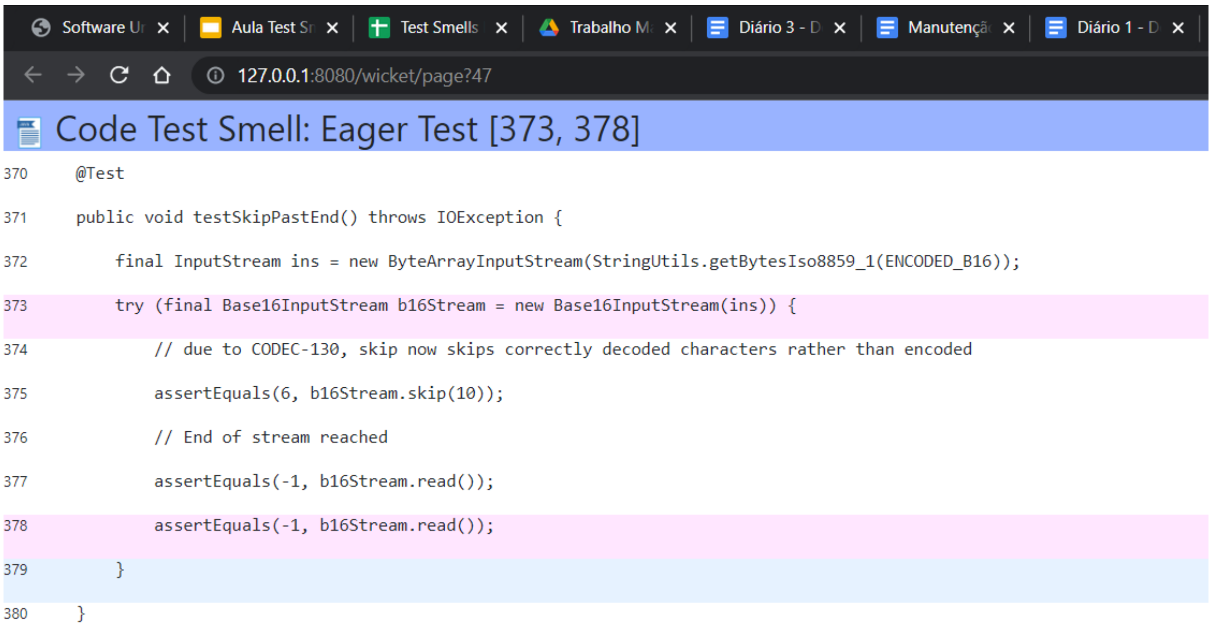
Teste refatorado



```
@Test
public void testSkipNone() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        final byte[] actualBytes = new byte[6];
        assertEquals(0, b16Stream.skip(0));
    }
}

public void testSkipNone() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        final byte[] actualBytes = new byte[6];
        b16Stream.read(actualBytes, 0, actualBytes.length);
        assertArrayEquals(actualBytes, new byte[] {(byte)202, (byte)254, (byte)186, (byte)190, (byte)255, (byte)255});
        // End of stream reached
        assertEquals(-1, b16Stream.read());
    }
}
```

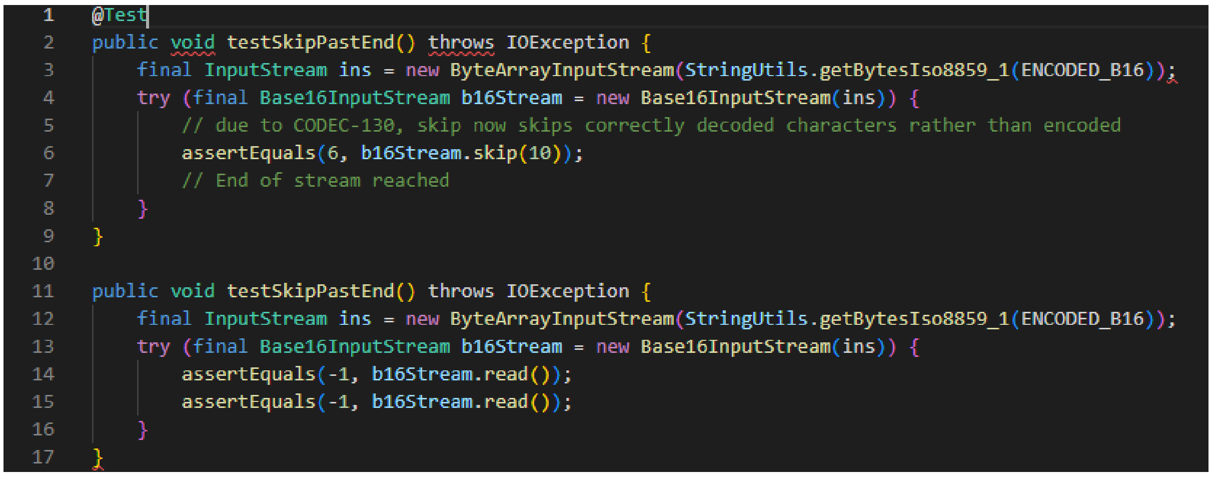
Eager Test - 3 (linha 373-378)



Code Test Smell: Eager Test [373, 378]

```
370    @Test
371    public void testSkipPastEnd() throws IOException {
372        final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
373        try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
374            // due to CODEC-130, skip now skips correctly decoded characters rather than encoded
375            assertEquals(6, b16Stream.skip(10));
376            // End of stream reached
377            assertEquals(-1, b16Stream.read());
378            assertEquals(-1, b16Stream.read());
379        }
380    }
```

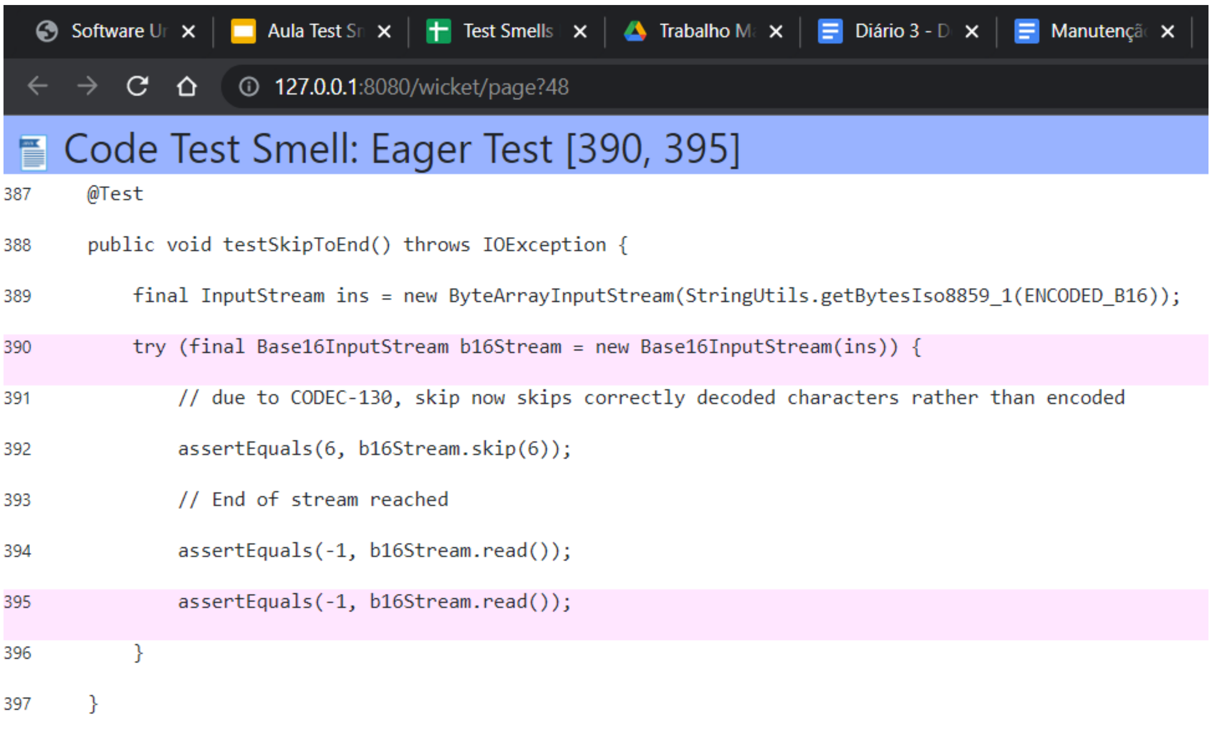
Teste refatorado



```
@Test
public void testSkipPastEnd() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        // due to CODEC-130, skip now skips correctly decoded characters rather than encoded
        assertEquals(6, b16Stream.skip(10));
        // End of stream reached
    }
}

public void testSkipPastEnd() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(-1, b16Stream.read());
        assertEquals(-1, b16Stream.read());
    }
}
```

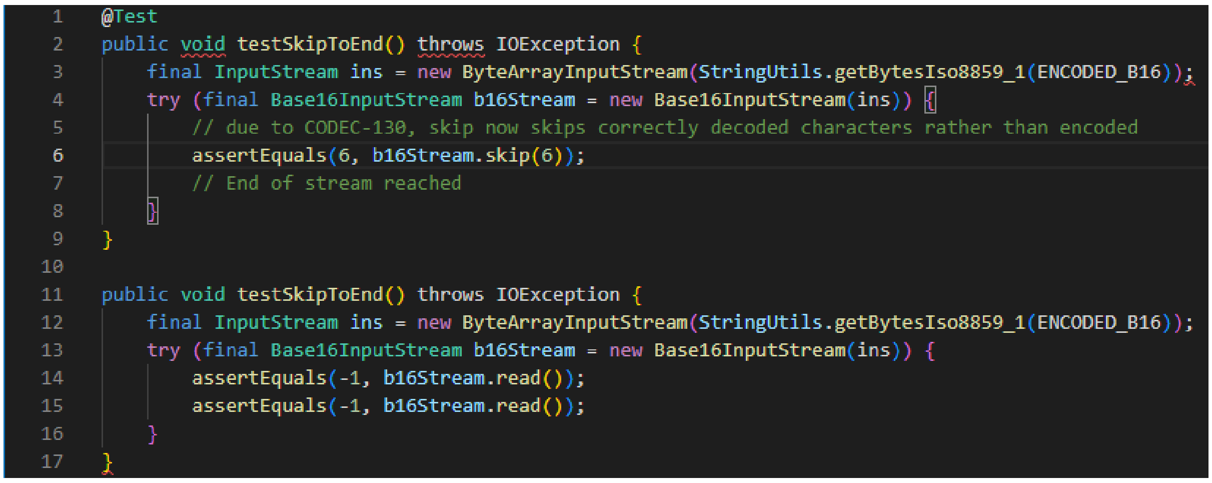
Eager Test - 4(linha 390-395)



Code Test Smell: Eager Test [390, 395]

```
387    @Test
388    public void testSkipToEnd() throws IOException {
389        final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
390        try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
391            // due to CODEC-130, skip now skips correctly decoded characters rather than encoded
392            assertEquals(6, b16Stream.skip(6));
393            // End of stream reached
394            assertEquals(-1, b16Stream.read());
395            assertEquals(-1, b16Stream.read());
396        }
397    }
```

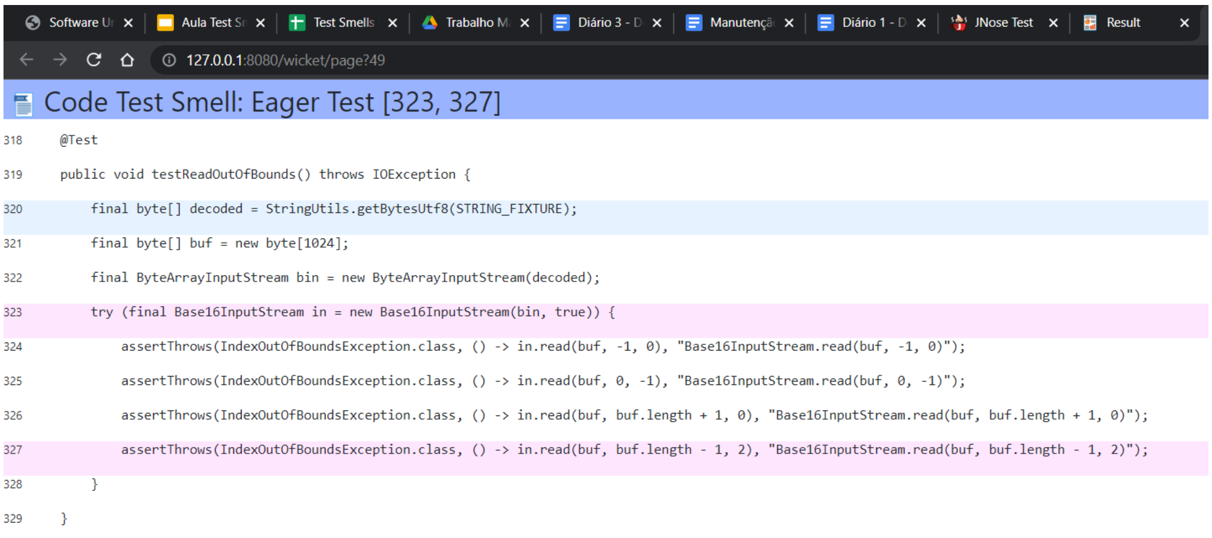
Teste refatorado



```
@Test
public void testSkipToEnd() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        // due to CODEC-130, skip now skips correctly decoded characters rather than encoded
        assertEquals(6, b16Stream.skip(6));
        // End of stream reached
    }
}

public void testSkipToEnd() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(-1, b16Stream.read());
        assertEquals(-1, b16Stream.read());
    }
}
```

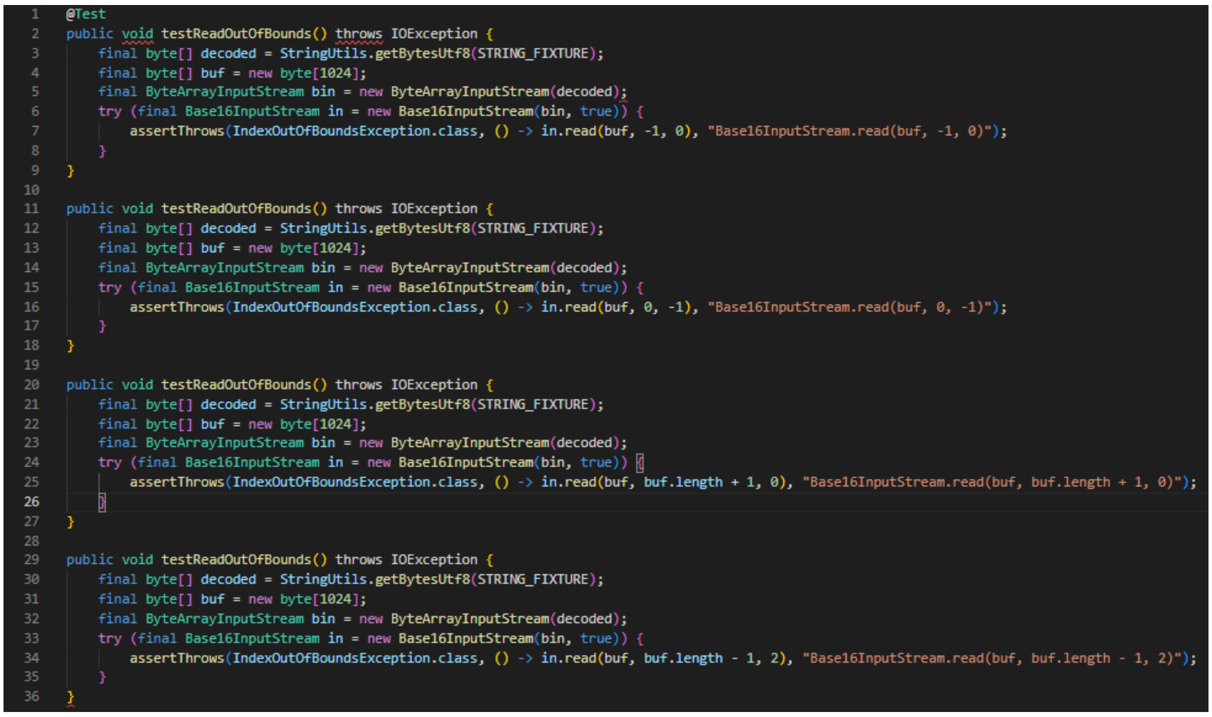
Eager Test - 5 (linha 323-327)



Code Test Smell: Eager Test [323, 327]

```
318     @Test
319     public void testReadOutOfBounds() throws IOException {
320         final byte[] decoded = StringUtils.getBytesUtf8(STRING_FIXTURE);
321         final byte[] buf = new byte[1024];
322         final ByteArrayInputStream bin = new ByteArrayInputStream(decoded);
323         try (final Base16InputStream in = new Base16InputStream(bin, true)) {
324             assertThrows(IndexOutOfBoundsException.class, () -> in.read(buf, -1, 0), "Base16InputStream.read(buf, -1, 0)");
325             assertThrows(IndexOutOfBoundsException.class, () -> in.read(buf, 0, -1), "Base16InputStream.read(buf, 0, -1)");
326             assertThrows(IndexOutOfBoundsException.class, () -> in.read(buf, buf.length + 1, 0), "Base16InputStream.read(buf, buf.length + 1, 0)");
327             assertThrows(IndexOutOfBoundsException.class, () -> in.read(buf, buf.length - 1, 2), "Base16InputStream.read(buf, buf.length - 1, 2)");
328         }
329     }
```

Teste refatorado



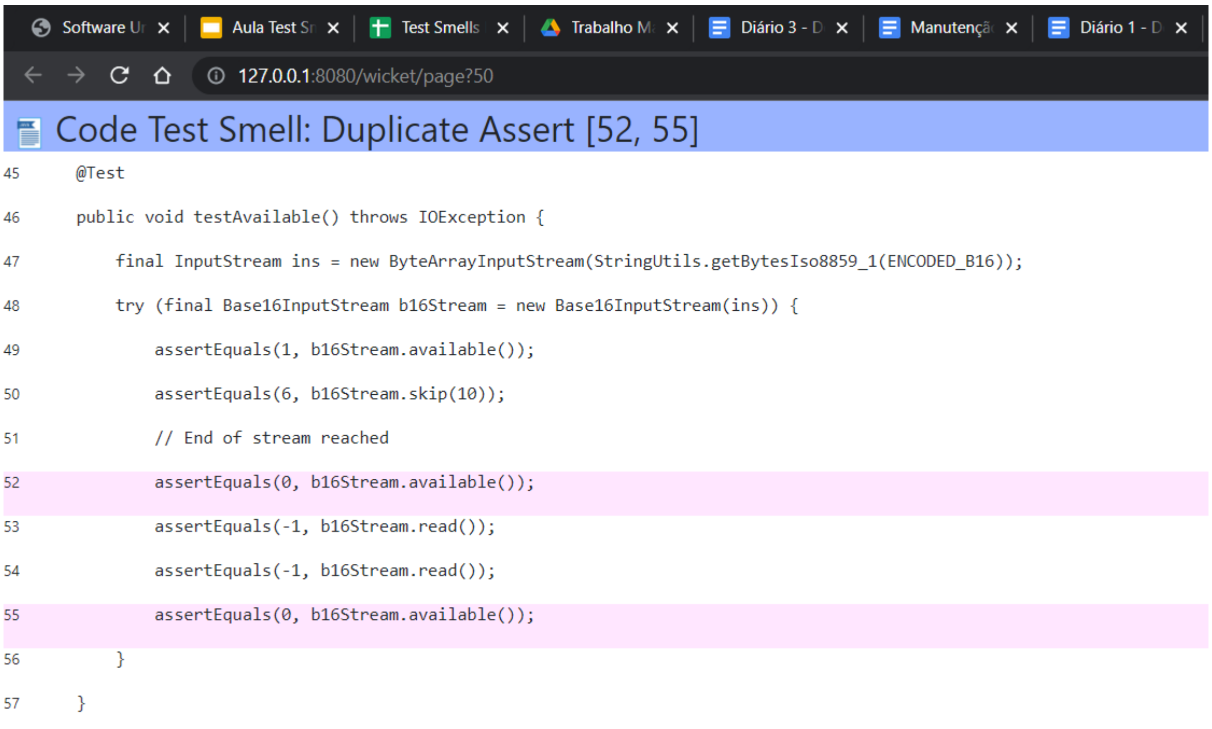
```java
@Test
public void testReadOutOfBounds() throws IOException {
    final byte[] decoded = StringUtils.getBytesUtf8(STRING_FIXTURE);
    final byte[] buf = new byte[1024];
    final ByteArrayInputStream bin = new ByteArrayInputStream(decoded);
    try (final Base16InputStream in = new Base16InputStream(bin, true)) {
        assertThrows(IndexOutOfBoundsException.class, () -> in.read(buf, -1, 0), "Base16InputStream.read(buf, -1, 0)");
    }
}

public void testReadOutOfBounds() throws IOException {
    final byte[] decoded = StringUtils.getBytesUtf8(STRING_FIXTURE);
    final byte[] buf = new byte[1024];
    final ByteArrayInputStream bin = new ByteArrayInputStream(decoded);
    try (final Base16InputStream in = new Base16InputStream(bin, true)) {
        assertThrows(IndexOutOfBoundsException.class, () -> in.read(buf, 0, -1), "Base16InputStream.read(buf, 0, -1)");
    }
}

public void testReadOutOfBounds() throws IOException {
    final byte[] decoded = StringUtils.getBytesUtf8(STRING_FIXTURE);
    final byte[] buf = new byte[1024];
    final ByteArrayInputStream bin = new ByteArrayInputStream(decoded);
    try (final Base16InputStream in = new Base16InputStream(bin, true)) {
        assertThrows(IndexOutOfBoundsException.class, () -> in.read(buf, buf.length + 1, 0), "Base16InputStream.read(buf, buf.length + 1, 0)");
    }
}

public void testReadOutOfBounds() throws IOException {
    final byte[] decoded = StringUtils.getBytesUtf8(STRING_FIXTURE);
    final byte[] buf = new byte[1024];
    final ByteArrayInputStream bin = new ByteArrayInputStream(decoded);
    try (final Base16InputStream in = new Base16InputStream(bin, true)) {
        assertThrows(IndexOutOfBoundsException.class, () -> in.read(buf, buf.length - 1, 2), "Base16InputStream.read(buf, buf.length - 1, 2)");
    }
}
```

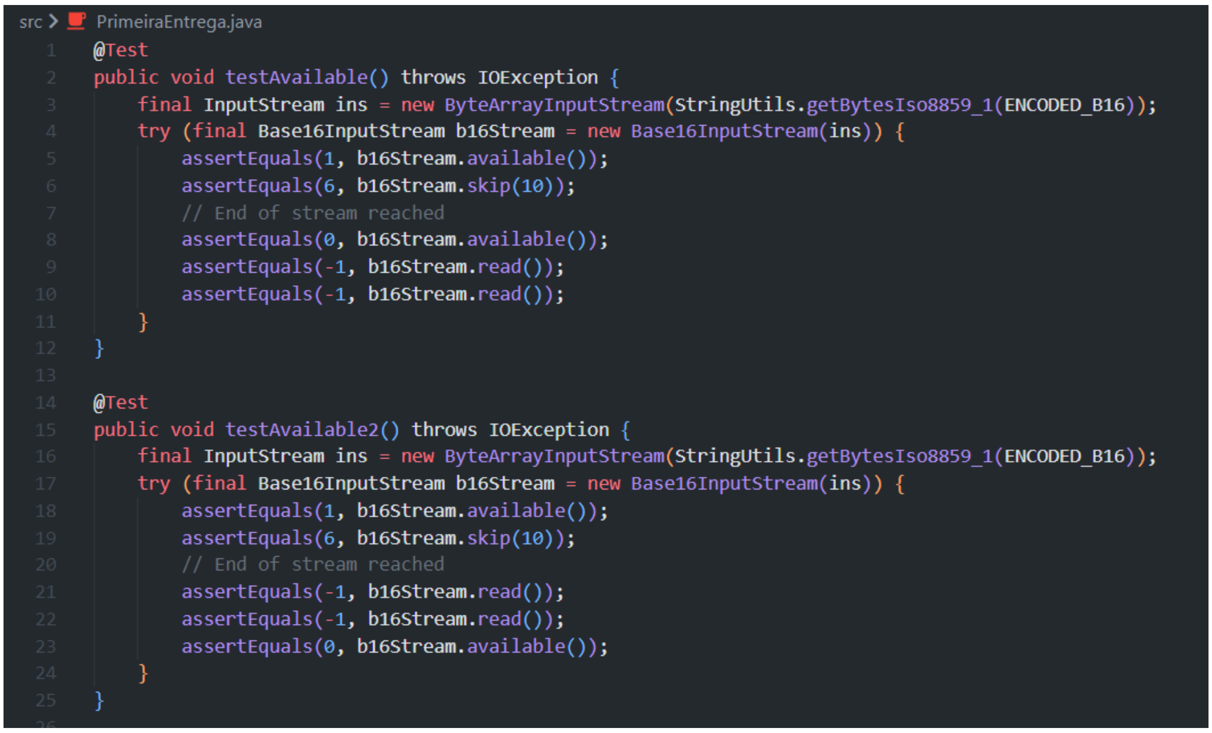
Duplicate Assertion - 1(linha 52-55)



Code Test Smell: Duplicate Assert [52, 55]

```
45      @Test
46      public void testAvailable() throws IOException {
47          final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
48          try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
49              assertEquals(1, b16Stream.available());
50              assertEquals(6, b16Stream.skip(10));
51              // End of stream reached
52              assertEquals(0, b16Stream.available());
53              assertEquals(-1, b16Stream.read());
54              assertEquals(-1, b16Stream.read());
55              assertEquals(0, b16Stream.available());
56          }
57      }
```

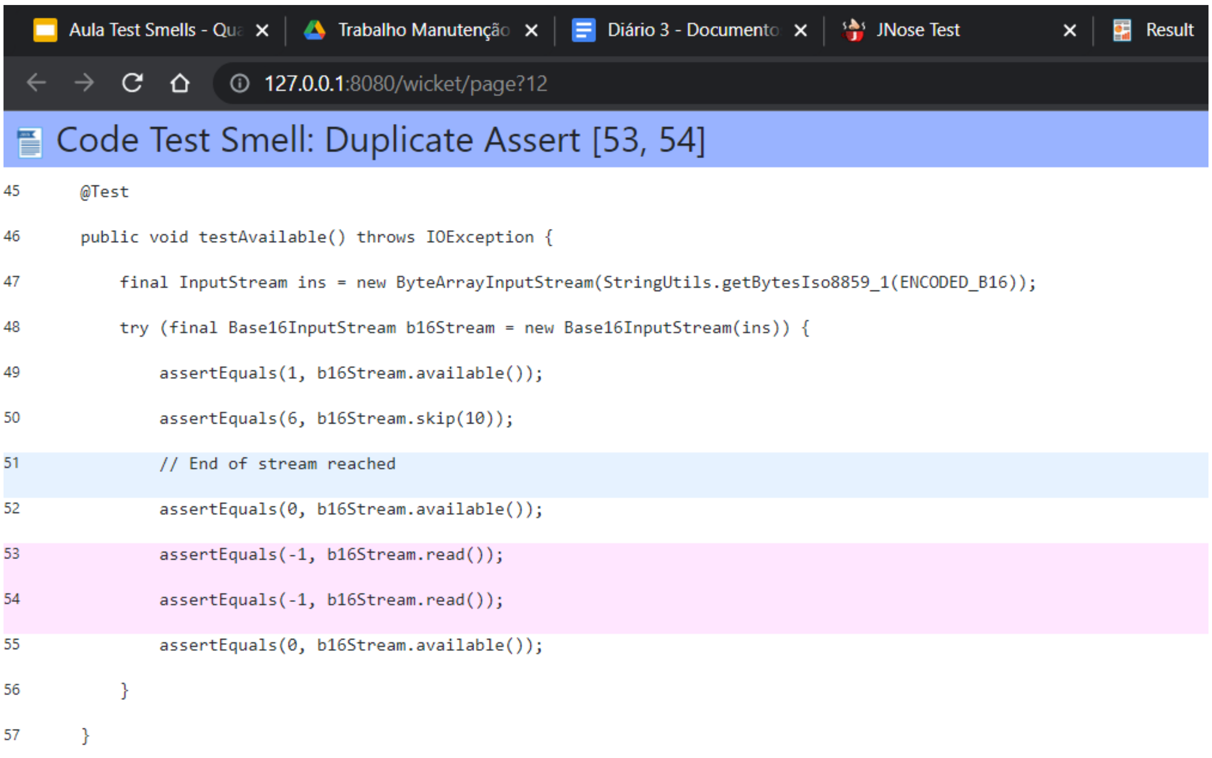
Teste refatorado



```java
@Test
public void testAvailable() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(1, b16Stream.available());
        assertEquals(6, b16Stream.skip(10));
        // End of stream reached
        assertEquals(0, b16Stream.available());
        assertEquals(-1, b16Stream.read());
        assertEquals(-1, b16Stream.read());
    }
}

@Test
public void testAvailable2() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(1, b16Stream.available());
        assertEquals(6, b16Stream.skip(10));
        // End of stream reached
        assertEquals(-1, b16Stream.read());
        assertEquals(-1, b16Stream.read());
        assertEquals(0, b16Stream.available());
    }
}
```

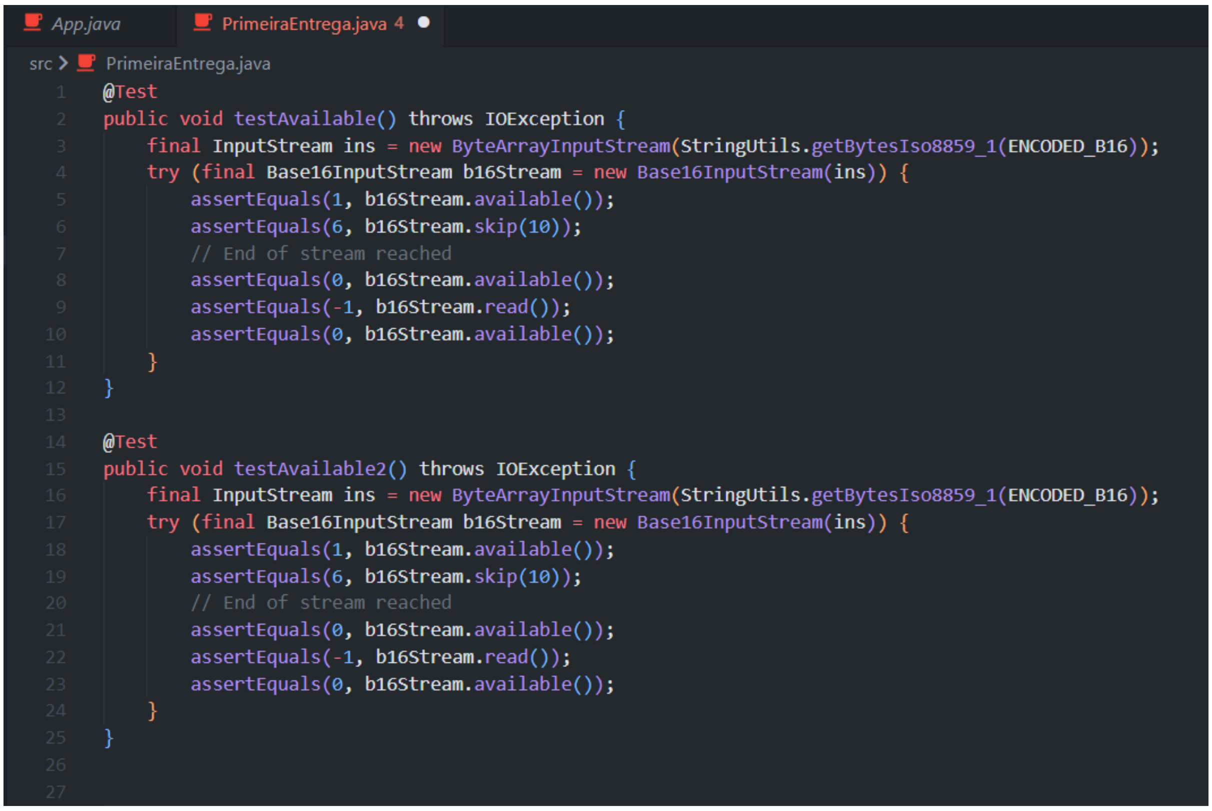
Duplicate Assertion - 2 (linha 53-54)



Code Test Smell: Duplicate Assert [53, 54]

```
45      @Test
46      public void testAvailable() throws IOException {
47          final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
48          try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
49              assertEquals(1, b16Stream.available());
50              assertEquals(6, b16Stream.skip(10));
51              // End of stream reached
52              assertEquals(0, b16Stream.available());
53              assertEquals(-1, b16Stream.read());
54              assertEquals(-1, b16Stream.read());
55              assertEquals(0, b16Stream.available());
56          }
57      }
```

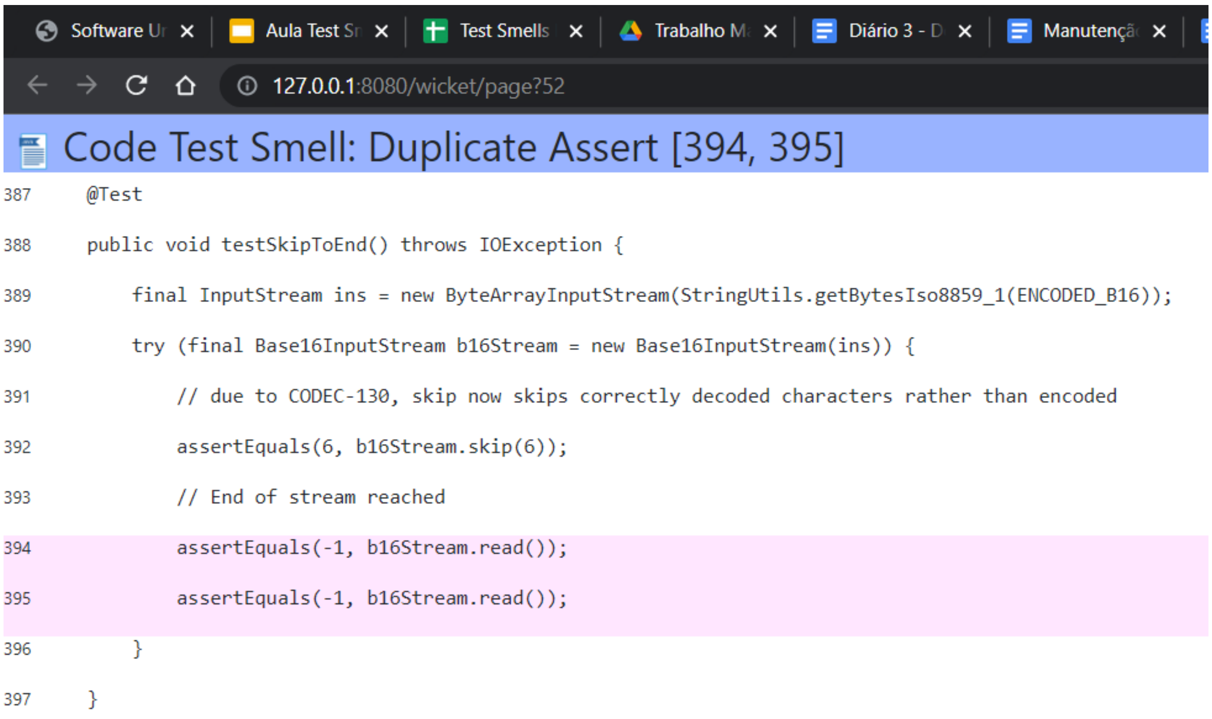
Teste Refatorado



```
@Test
public void testAvailable() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(1, b16Stream.available());
        assertEquals(6, b16Stream.skip(10));
        // End of stream reached
        assertEquals(0, b16Stream.available());
        assertEquals(-1, b16Stream.read());
        assertEquals(0, b16Stream.available());
    }
}

@Test
public void testAvailable2() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(1, b16Stream.available());
        assertEquals(6, b16Stream.skip(10));
        // End of stream reached
        assertEquals(0, b16Stream.available());
        assertEquals(-1, b16Stream.read());
        assertEquals(0, b16Stream.available());
    }
}
```

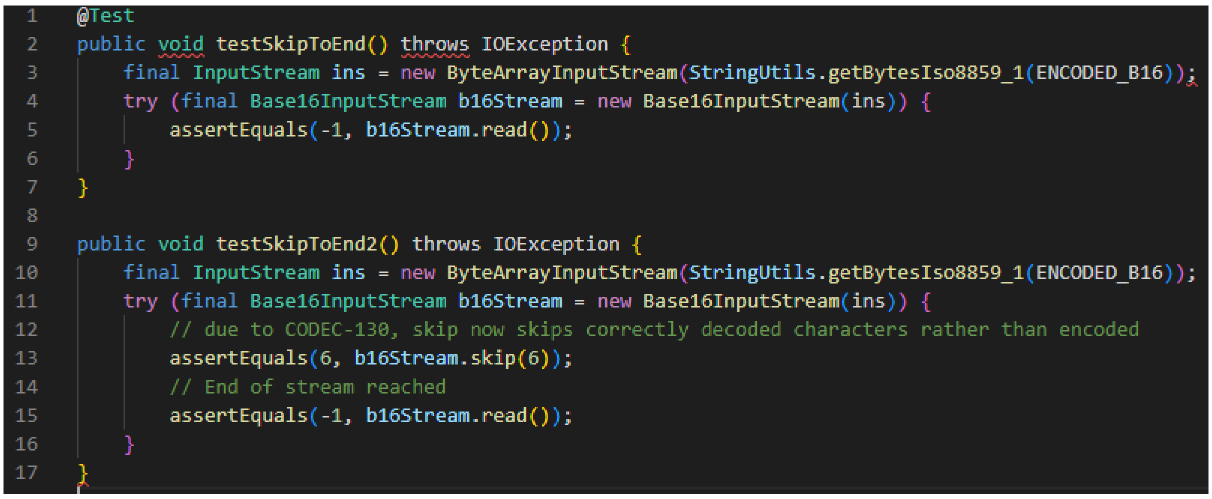
Duplicate Assertion - 3 (linha 394-395)



Code Test Smell: Duplicate Assert [394, 395]

```
387    @Test
388    public void testSkipToEnd() throws IOException {
389        final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
390        try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
391            // due to CODEC-130, skip now skips correctly decoded characters rather than encoded
392            assertEquals(6, b16Stream.skip(6));
393            // End of stream reached
394            assertEquals(-1, b16Stream.read());
395            assertEquals(-1, b16Stream.read());
396        }
397    }
```

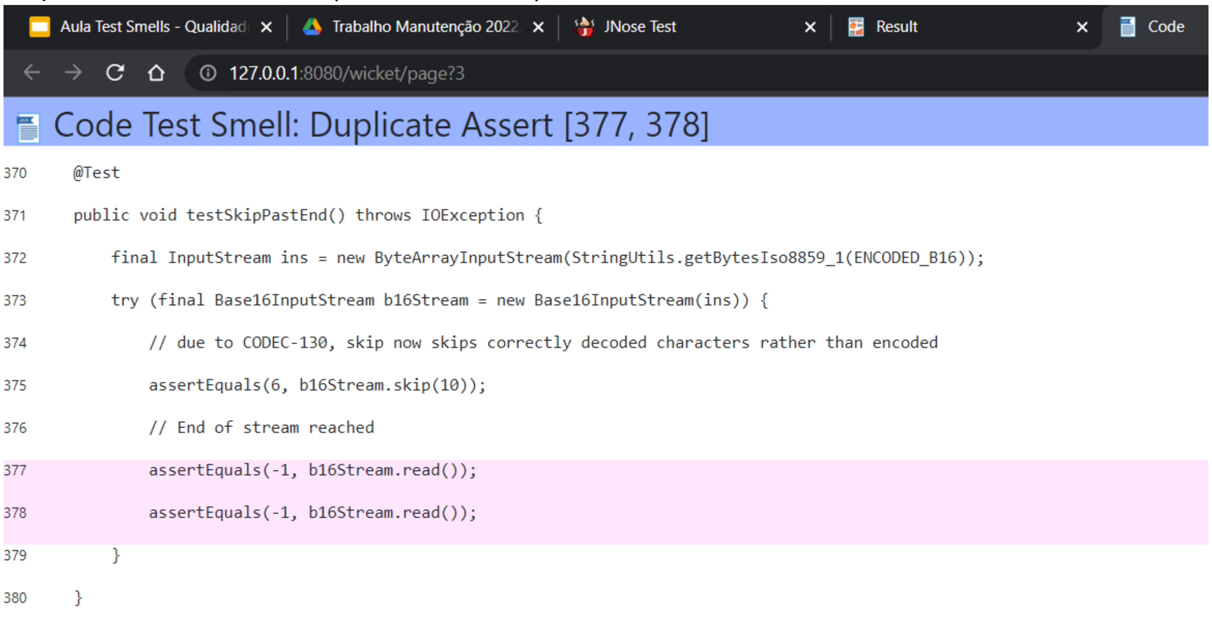
Teste refatorado



```
@Test
public void testSkipToEnd() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(-1, b16Stream.read());
    }
}

public void testSkipToEnd2() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        // due to CODEC-130, skip now skips correctly decoded characters rather than encoded
        assertEquals(6, b16Stream.skip(6));
        // End of stream reached
        assertEquals(-1, b16Stream.read());
    }
}
```

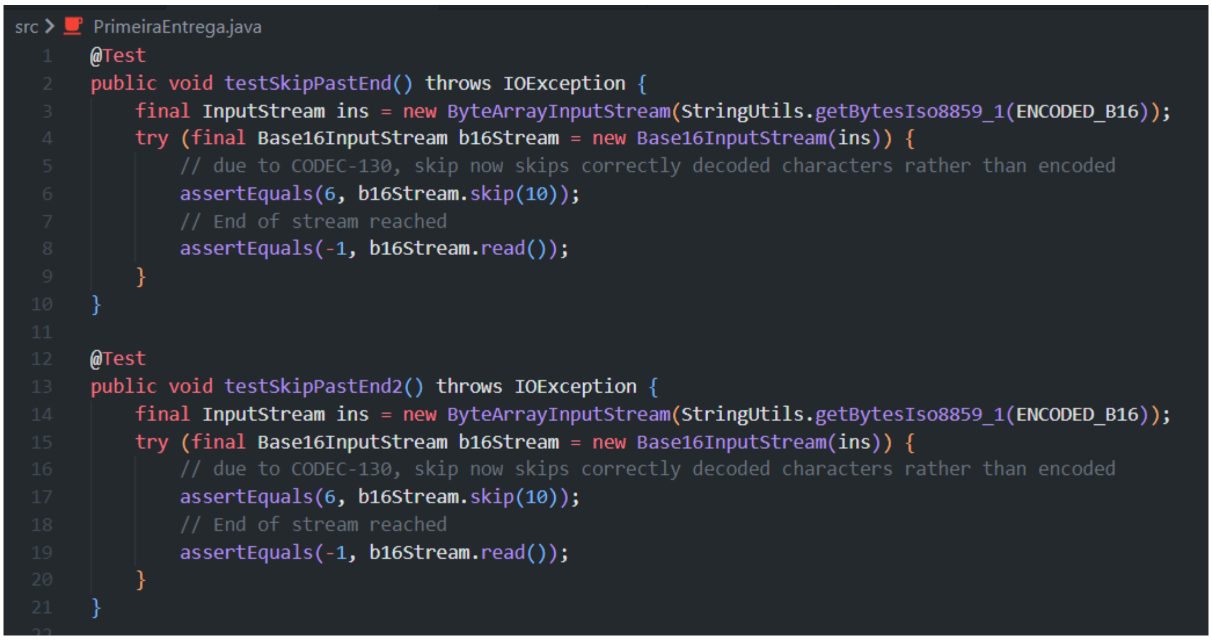
Duplicate Assertion - 4 (linha 377-378)



Code Test Smell: Duplicate Assert [377, 378]

```
370  @Test
371  public void testSkipPastEnd() throws IOException {
372      final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
373      try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
374          // due to CODEC-130, skip now skips correctly decoded characters rather than encoded
375          assertEquals(6, b16Stream.skip(10));
376          // End of stream reached
377          assertEquals(-1, b16Stream.read());
378          assertEquals(-1, b16Stream.read());
379      }
380  }
```

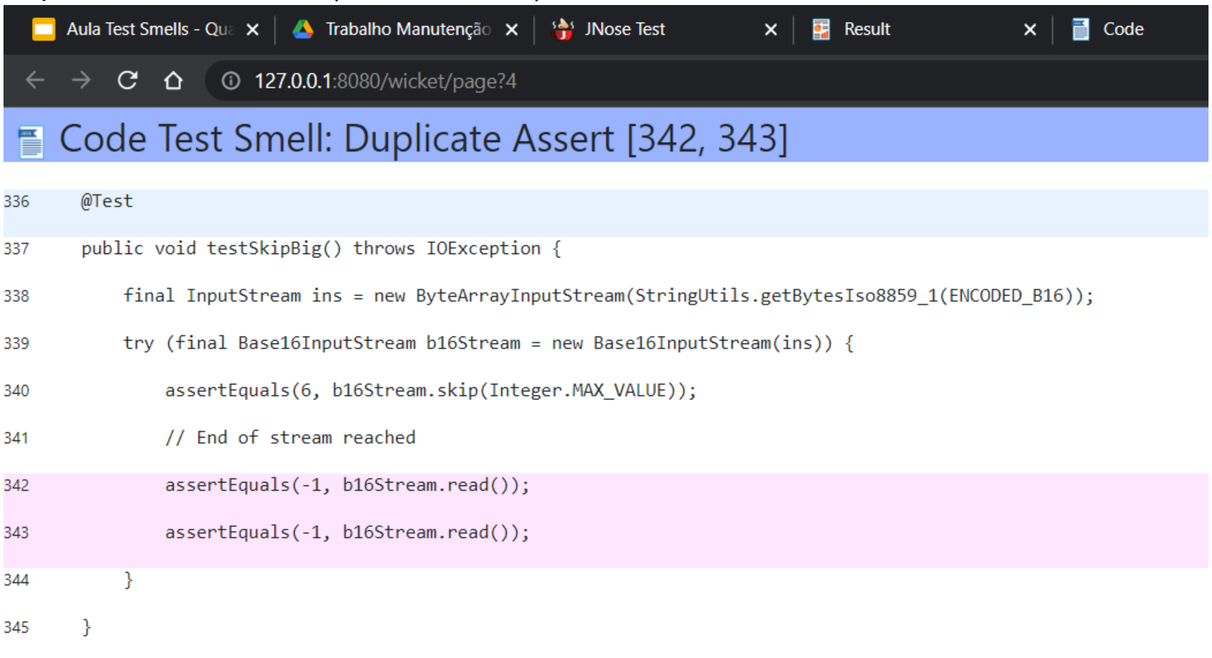
Teste refatorado



```
@Test
public void testSkipPastEnd() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        // due to CODEC-130, skip now skips correctly decoded characters rather than encoded
        assertEquals(6, b16Stream.skip(10));
        // End of stream reached
        assertEquals(-1, b16Stream.read());
    }
}

@Test
public void testSkipPastEnd2() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        // due to CODEC-130, skip now skips correctly decoded characters rather than encoded
        assertEquals(6, b16Stream.skip(10));
        // End of stream reached
        assertEquals(-1, b16Stream.read());
    }
}
```

Duplicate Assertion - 5 (linha 342-343)



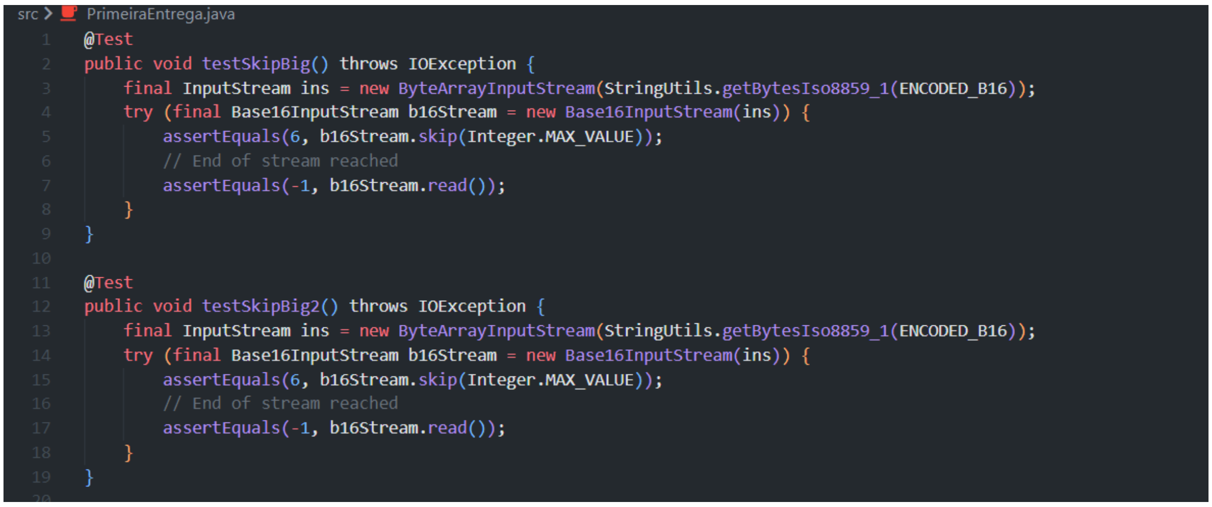
Aula Test Smells - Qu | Trabalho Manutenção | JNose Test | Result | Code

127.0.0.1:8080/wicket/page?4

Code Test Smell: Duplicate Assert [342, 343]

```
336     @Test
337     public void testSkipBig() throws IOException {
338         final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
339         try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
340             assertEquals(6, b16Stream.skip(Integer.MAX_VALUE));
341             // End of stream reached
342             assertEquals(-1, b16Stream.read());
343             assertEquals(-1, b16Stream.read());
344         }
345     }
```

Teste refatorado:



src > PrimeiraEntrega.java

```
@Test
public void testSkipBig() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(6, b16Stream.skip(Integer.MAX_VALUE));
        // End of stream reached
        assertEquals(-1, b16Stream.read());
    }
}

@Test
public void testSkipBig2() throws IOException {
    final InputStream ins = new ByteArrayInputStream(StringUtils.getBytesIso8859_1(ENCODED_B16));
    try (final Base16InputStream b16Stream = new Base16InputStream(ins)) {
        assertEquals(6, b16Stream.skip(Integer.MAX_VALUE));
        // End of stream reached
        assertEquals(-1, b16Stream.read());
    }
}
```