

## Testes de Unidade



Universidade de Brasília

Faculdade UnB Gama 

FGA 0238 Testes de Software - Turma 02

Nome Nicolas Chagas Souza

Equipe PassaNada

Semestre

2023.1

Matrícula

20/0042327

### Atividade 3 – Desenvolver Testes de Unidade

## 1. Identificação do Projeto

A aplicação em teste é o TEAMMATES, que fornece funcionalidades para avaliação entre pares. Para utilizar o sistema, um instrutor deve solicitar uma conta na plataforma, em seguida, ele pode criar cursos, inscrever alunos nesses cursos, separados por grupos, e gerar sessões de avaliações por pares, onde cada integrante avalia a si mesmo e aos demais integrantes do grupo, por meio de questões definidas pelo instrutor.

O fork utilizado está disponível no [link](#).

## 2. Cobertura de Testes

A execução dos testes de backend gerou o relatório representado nas figuras seguintes.



### teammates

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
<a href="#">teammates.ui.webapi</a>		87%		79%	245	1,194	470	3,923	33	572	10	124
<a href="#">teammates.logic.external</a>		19%		8%	183	208	383	464	55	75	4	14
<a href="#">teammates.common.util</a>		70%		61%	179	499	309	1,075	53	252	19	44
<a href="#">teammates.ui.constants</a>		0%		n/a	15	15	123	123	15	15	4	4
<a href="#">teammates.ui.servlets</a>		31%		27%	96	127	318	451	44	62	11	15
<a href="#">teammates.logic.api</a>		83%		61%	204	500	81	1,083	15	237	0	8
<a href="#">teammates.storage.search</a>		24%		20%	50	82	150	220	25	57	5	10

<a href="#">teammates.storage.search</a>		24%		30%	59	82	158	238	35	57	5	10
<a href="#">teammates.logic.core</a>		92%		85%	143	883	149	2,152	13	346	0	15
<a href="#">teammates.storage.api</a>		90%		82%	142	687	72	1,388	12	298	1	15
<a href="#">teammates.ui.output</a>		85%		67%	136	498	161	1,089	79	393	3	66
<a href="#">teammates.common.datatransfer.attributes</a>		95%		78%	142	1,050	109	1,980	23	748	0	62
<a href="#">teammates.common.datatransfer.questions</a>		92%		88%	115	583	74	1,011	50	263	0	25
<a href="#">teammates.common.datatransfer</a>		93%		81%	71	436	40	932	10	228	0	20
<a href="#">teammates.storage.entity</a>		86%		74%	41	325	68	620	29	300	0	15
<a href="#">teammates.ui.request</a>		89%		63%	72	295	28	418	3	181	0	32
<a href="#">teammates.common.datatransfer.logs</a>		82%		37%	37	152	42	266	31	144	1	15
<a href="#">teammates.main</a>		0%		0%	9	9	27	27	8	8	2	2
<a href="#">teammates.common.exception</a>		52%		n/a	7	16	15	33	7	16	4	10
Total	12,563 of 72,954	82%	1,729 of 6,586	73%	1,896	7,559	2,627	17,273	515	4,195	64	496

Created with [JaCoCo](#) 0.8.8.202204050719

Figura 1. Relatório geral dos testes. (Fonte: elaborado pelo autor, 2023)

[teammates](#) > [teammates.common.util](#) [Source Files](#) [Sessi](#)

## teammates.common.util

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
<a href="#">TzdbResourceZoneRulesProvider</a>		0%		0%	41	41	130	130	12	12	1	1
<a href="#">Logger</a>		47%		35%	26	44	85	162	8	23	0	1
<a href="#">Config</a>		71%		18%	32	42	20	86	5	13	0	1
<a href="#">FieldValidator</a>		93%		88%	24	155	2	239	0	52	0	1
<a href="#">RequestTracer</a>		48%		40%	7	12	10	23	2	7	0	1
<a href="#">HttpRequest</a>		0%		0%	2	2	11	11	1	1	1	1
<a href="#">HttpRequestHelper</a>		77%		33%	9	19	5	31	0	10	0	1
<a href="#">StringHelper</a>		91%		96%	1	36	9	70	0	23	0	1
<a href="#">JsonUtils.ZoneldAdapter</a>		0%		n/a	2	2	4	4	2	2	1	1
<a href="#">RequestTracer.RequestTrace</a>		0%		n/a	1	1	6	6	1	1	1	1
<a href="#">TimeHelper</a>		90%		79%	7	26	5	32	0	9	0	1

Figura 2. Relatório das classes do pacote util. (Fonte: elaborado pelo autor, 2023)

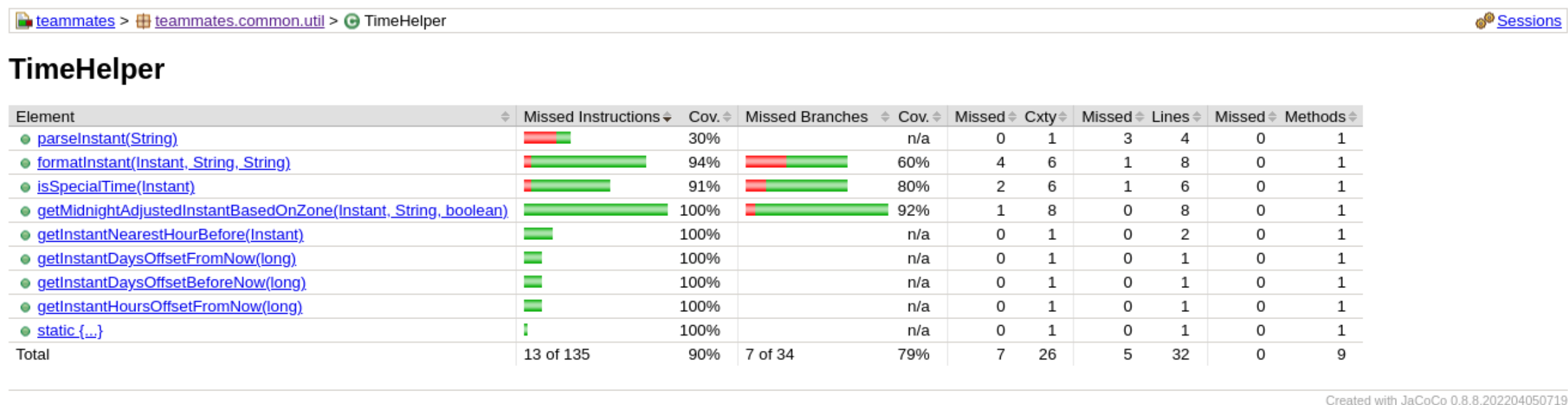


Figura 3. Relatório dos métodos da classe TimeHelper. (Fonte: elaborado pelo autor, 2023)

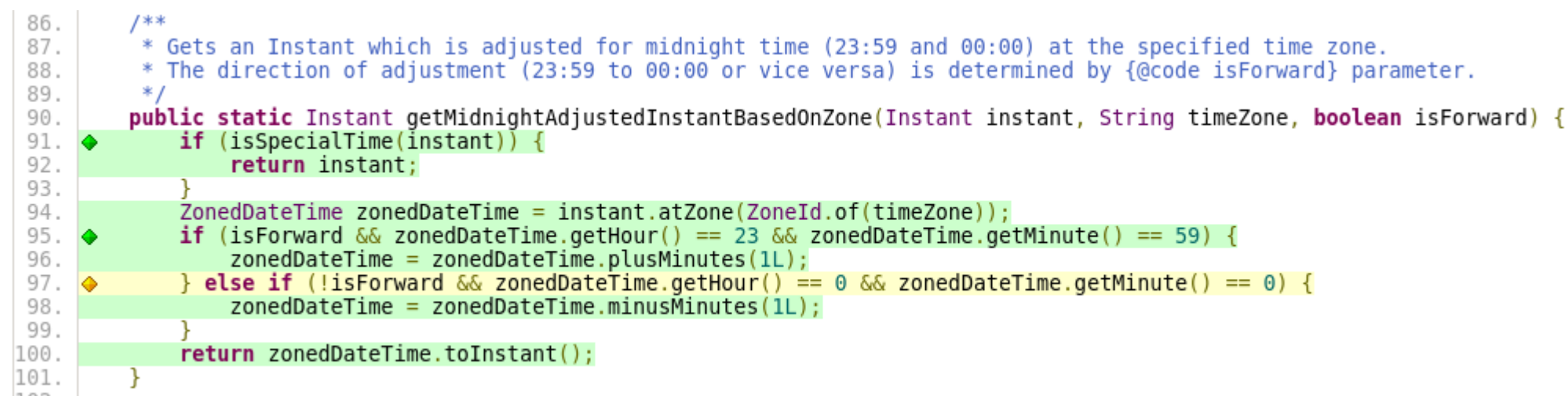


Figura 4. Cobertura do método testado antes da criação dos testes unitários. (Fonte: elaborado pelo autor, 2023)

3. Método testado

O método escolhido para teste foi o `getMidnightAdjustedInstantBaseOnZone` da classe `teammates.common.util.TimeHelper`.



```

9      ZonedDateTime zonedDateTime = instant.atZone(ZoneId.of(timeZone));
10     if (isForward && zonedDateTime.getHour() == 23 && zonedDateTime.getMinute() == 59) {
11         zonedDateTime = zonedDateTime.plusMinutes(1L);
12     } else if (!isForward && zonedDateTime.getHour() == 0 && zonedDateTime.getMinute() == 0) {
13         zonedDateTime = zonedDateTime.minusMinutes(1L);
14     }
15     return zonedDateTime.toInstant();
16 }

```

## 4. Classe de Testes

Antes da incrementação dos testes existentes, o método de teste da classe estava no seguinte estado:

**teammates.common.util.TimeHelperTest.java**

```

1      @Test
2      public void testGetMidnightAdjustedInstantBasedOnZone() {
3          String zoneId = "UTC";
4          Instant instantAt0000 = LocalDateTime.of(2015, Month.NOVEMBER, 30, 0, 0).atZone(ZoneId.of(zoneId)).toInstant();
5
6          Instant backwardAdjusted = TimeHelper.getMidnightAdjustedInstantBasedOnZone(instantAt0000, zoneId, false);
7          assertEquals("Sun, 29 Nov 2015, 11:59 PM UTC",
8              TimeHelper.formatInstant(backwardAdjusted, zoneId, DATETIME_DISPLAY_FORMAT));
9
10         Instant forwardAdjusted = TimeHelper.getMidnightAdjustedInstantBasedOnZone(instantAt0000, zoneId, true);
11         assertEquals("Mon, 30 Nov 2015, 12:00 AM UTC",
12             TimeHelper.formatInstant(forwardAdjusted, zoneId, DATETIME_DISPLAY_FORMAT));
13
14         Instant instantAt2359 = LocalDateTime.of(2015, Month.NOVEMBER, 29, 23, 59).atZone(ZoneId.of(zoneId)).toInstant();
15
16         backwardAdjusted = TimeHelper.getMidnightAdjustedInstantBasedOnZone(instantAt2359, zoneId, false);
17         assertEquals("Sun, 29 Nov 2015, 11:59 PM UTC",
18             TimeHelper.formatInstant(backwardAdjusted, zoneId, DATETIME_DISPLAY_FORMAT));
19
20         forwardAdjusted = TimeHelper.getMidnightAdjustedInstantBasedOnZone(instantAt2359, zoneId, true);
21         assertEquals("Mon, 30 Nov 2015, 12:00 AM UTC",
22             TimeHelper.formatInstant(forwardAdjusted, zoneId, DATETIME_DISPLAY_FORMAT));
23
24         String wrongTimeZone = "Asia/Singapore";
25
26         backwardAdjusted = TimeHelper.getMidnightAdjustedInstantBasedOnZone(instantAt0000, wrongTimeZone, false);
27         assertEquals("Mon, 30 Nov 2015, 12:00 AM UTC",
28             TimeHelper.formatInstant(backwardAdjusted, zoneId, DATETIME_DISPLAY_FORMAT));
29
30         forwardAdjusted = TimeHelper.getMidnightAdjustedInstantBasedOnZone(instantAt2359, wrongTimeZone, true);
31         assertEquals("Sun, 29 Nov 2015, 11:59 PM UTC",
32             TimeHelper.formatInstant(forwardAdjusted, zoneId, DATETIME_DISPLAY_FORMAT));
33     }

```

## 5. Tabela de decisões/condições

Decisão	Situação para <code>true</code>	Situação para <code>false</code>
3	O <code>instant</code> é um entre: <code>Const.TIME_REPRESENTS_FOLLOW_OPENING</code> <code>Const.TIME_REPRESENTS_FOLLOW_VISIBLE</code> <code>Const.TIME_REPRESENTS_LATER</code> <code>Const.TIME_REPRESENTS_NOW</code>	O <code>instant</code> não é uma dessas constantes.
7	A formatação do tempo é para frente ( <code>isForward == true</code> ) e o horário é 23:59	A formatação do tempo não é para frente ( <code>isForward == false</code> ) ou o horário é diferente de 23:59.
9	A formatação do tempo não é para frente ( <code>isForward == false</code> ) e o horário é 00:00	A formatação do tempo é para frente ( <code>isForward == true</code> ) ou o horário é diferente de 00:00.

Tabela 1. Identificação das decisões. (Fonte: elaborado pelo autor, 2023)

Decisão	Condição	Situação para <code>true</code>	Situação para <code>false</code>
3	<code>isSpecialTime(instant)</code>	<code>instant</code> é um entre: <code>Const.TIME_REPRESENTS_FOLLOW_OPENING</code> <code>Const.TIME_REPRESENTS_FOLLOW_VISIBLE</code> <code>Const.TIME_REPRESENTS_LATER</code> <code>Const.TIME_REPRESENTS_NOW</code> (1)	<code>instant</code> é diferente desses. (2)
7A	<code>isForward</code>	<code>isForward==true</code> (3)	<code>isForward==false</code> (4)
7B	<code>zonedDateTime.getHour() == 23</code>	O <code>instant</code> está entre 23:00 e 23:59. (5)	O <code>instant</code> está fora do intervalo de 23:00 à 23:59. (6)
7C	<code>zonedDateTime.getMinute() == 59</code>	O <code>instant</code> é do tipo <code>hh:59</code> , com <code>hh</code> entre 0 e 23. (7)	O <code>instant</code> é do tipo <code>hh:mm</code> , com <code>mm != 59</code> . (8)
9A	<code>!isForward</code>	<code>isForward==false</code> (9)	<code>isForward==true</code> (10)
9B	<code>zonedDateTime.getHour() == 0</code>	O <code>instant</code> está entre 00:00 e 00:59. (11)	O <code>instant</code> está fora do intervalo de 00:00 à 00:59. (12)
9C	<code>zonedDateTime.getMinute() == 0</code>	O <code>instant</code> é do tipo <code>hh:00</code> , com <code>hh</code> entre 0 e 23. (13)	O <code>instant</code> é do tipo <code>hh:mm</code> , com <code>mm != 00</code> . (14)

Tabela 2. Identificação das condições. (Fonte: elaborado pelo autor, 2023)

6. Especificação dos Casos de Teste

A técnica das combinações múltiplas foi utilizadas para a determinação dos casos de teste, representados na tabela a seguir. Observe que 7A e 9A são opostas, então não é possível 7A == 9A. As condições 7B e 9B são incompatíveis, então não é possível ter 7B == 9B. As condições 7C e 9C são incompatíveis, então não é possível ter 7C == 9B. Como todas as constantes especiais de tempo possuem horário `00:00`, a condição 3 ser verdadeira implica em 9B e 9C verdadeiros e 7B e 7C falsos. Levando em consideração esses fatores, as combinações possíveis entre as decisões 7A, 7B e 7C e as decisões 9A, 9B e 9C foram representadas na tabela a seguir.

Número da combinação	3	7A	7B	7C	9A	9B	9C	Caso de Teste
#1	X	F	F	F	T	T	T	3
#2	F	F	F	T	T	T	F	4
#3	F	F	T	F	T	F	T	5
#4	F	F	T	T	T	F	F	2
#5	X	T	F	F	F	T	T	1
#6	F	T	F	T	F	T	F	6
#7	F	T	T	F	F	F	T	7
#8	F	T	T	T	F	F	F	8

Tabela 3. Combinações entre as condições. (Fonte: elaborado pelo autor, 2023)

X denota o 'don't care'.

Número	Condições	Condição de Entrada			Combinações	Saída Esperada
		<code>instant</code>	<code>timeZone</code>	<code>isForward</code>	<code>  3   7A   7B   7C   9A   9B   9C  </code>	
1	1, 3, 6, 8, 10, 11, 13	<code>TIME_REPRESENTS_FOLLOW_OPENING¹</code>	GMT-3	true	<code>  T   T   F   F   F   T   T  </code>	TIME_REPRESENTES_FOLLOW_OPENING
2	2, 4, 5, 7, 9, 12, 14	2023-05-31 23:59:00	GMT-3	false	<code>  F   F   T   T   T   F   F  </code>	2023-05-31 23:59:00
3	-	2023-05-31 00:00:00	GMT-3	false	<code>  F   F   F   F   T   T   T  </code>	2023-05-30 23:59:00
4	-	2023-05-31 00:59:00	GMT-3	false	<code>  F   F   F   T   T   T   F  </code>	2023-05-31 00:59:00
5	-	2023-05-31 23:00:00	GMT-3	false	<code>  F   F   T   F   T   F   T  </code>	2023-05-31 23:00:00

6	-	2023-05-31 00:59:00	GMT-3	true	F   T   F   T   F   T   F	2023-05-31 00:59:00
7	-	2023-05-31 23:00:00	GMT-3	true	F   T   T   F   F   F   T	2023-05-31 23:00:00
8	-	2023-05-31 23:59:00	GMT-3	true	F   T   T   T   F   F   F	2023-06-01 00:00:00

<sup>1</sup> Const.TIME\_REPRESENTS\_FOLLOW\_OPENING = TimeHelper.parseInstant("1970-12-31 00:00:00Z") como o horário dessa constante é 0:00, ela já satisfaz as condições 11 e 13.

## 7. Implementação dos Casos de Teste

TimeHelperTest.java

```
1 // Test Case #1
2 @Test
3 public void testGetMidnightAdjustedInstantBasedOnZoneWithProjectsTimeConstantShouldReturnSameConstant() {
4     String timeZone = "GMT-3";
5     Instant constantInput = Const.TIME_REPRESENTS_FOLLOW_OPENING;
6     assertNotNull(constantInput);
7     Instant adjustedTime = TimeHelper.getMidnightAdjustedInstantBasedOnZone(constantInput, timeZone, true);
8     assertEquals(constantInput, adjustedTime);
9 }
10
11 // Test Case #2
12 @Test
13 public void testGetMidnightAdjustedInstantBasedOnZoneWithTime2359notForwardFalseShouldReturnSameInstant() {
14     String timeZone = "GMT-3";
15     Instant instantAt2359 = LocalDateTime.of(2023, Month.MAY, 31, 23, 59).atZone(ZoneId.of(timeZone)).toInstant();
16     Instant adjustedTime = TimeHelper.getMidnightAdjustedInstantBasedOnZone(instantAt2359, timeZone, false);
17     assertEquals(instantAt2359, adjustedTime);
18 }
19
20 // Test Case #3
21 @Test
22 public void testGetMidnightAdjustedInstantBasedOnZoneWithTime0000notForwardShouldReturnPreviousDayAt2359() {
23     String timeZone = "GMT-3";
24     Instant instantAt0000 = LocalDateTime.of(2023, Month.MAY, 31, 0, 0).atZone(ZoneId.of(timeZone)).toInstant();
25     Instant previousDayAt2359 = LocalDateTime.of(2023, Month.MAY, 30, 23, 59).atZone(ZoneId.of(timeZone))
26         .toInstant();
27     Instant adjustedTime = TimeHelper.getMidnightAdjustedInstantBasedOnZone(instantAt0000, timeZone, false);
28     assertEquals(previousDayAt2359, adjustedTime);
29 }
30
31 // Test Case #4
32 @Test
33 public void testGetMidnightAdjustedInstantBasedOnZoneWithTime0059notForwardShouldReturnSameInstant() {
34     String timeZone = "GMT-3";
35     Instant instantAt0059 = LocalDateTime.of(2023, Month.MAY, 31, 0, 59).atZone(ZoneId.of(timeZone)).toInstant();
36     Instant adjustedTime = TimeHelper.getMidnightAdjustedInstantBasedOnZone(instantAt0059, timeZone, false);
37     assertEquals(instantAt0059, adjustedTime);
38 }
39
```



```

40 // Test Case #5
41 @Test
42 public void testGetMidnightAdjustedInstantBasedOnZoneWithTime2300notForwardShouldReturnSameInstant() {
43     String timeZone = "GMT-3";
44     Instant instantAt2300 = LocalDateTime.of(2023, Month.MAY, 31, 23, 0).atZone(ZoneId.of(timeZone)).toInstant();
45     Instant adjustedTime = TimeHelper.getMidnightAdjustedInstantBasedOnZone(instantAt2300, timeZone, false);
46     assertEquals(instantAt2300, adjustedTime);
47 }
48
49 // Test Case #6
50 @Test
51 public void testGetMidnightAdjustedInstantBasedOnZoneWithTime0059andForwardShouldReturnSameInstant() {
52     String timeZone = "GMT-3";
53     Instant instantAt0059 = LocalDateTime.of(2023, Month.MAY, 31, 0, 59).atZone(ZoneId.of(timeZone)).toInstant();
54     Instant adjustedTime = TimeHelper.getMidnightAdjustedInstantBasedOnZone(instantAt0059, timeZone, true);
55     assertEquals(instantAt0059, adjustedTime);
56 }
57
58 // Test Case #7
59 @Test
60 public void testGetMidnightAdjustedInstantBasedOnZoneWithTime2300andForwardShouldReturnSameInstant() {
61     String timeZone = "GMT-3";
62     Instant instantAt2300 = LocalDateTime.of(2023, Month.MAY, 31, 23, 0).atZone(ZoneId.of(timeZone)).toInstant();
63     Instant adjustedTime = TimeHelper.getMidnightAdjustedInstantBasedOnZone(instantAt2300, timeZone, true);
64     assertEquals(instantAt2300, adjustedTime);
65 }
66
67 // Test Case #8
68 @Test
69 public void testGetMidnightAdjustedInstantBasedOnZoneWithTime2359andForwardShouldReturnNextDayAt0000() {
70     String timeZone = "GMT-3";
71     Instant instantAt2359 = LocalDateTime.of(2023, Month.MAY, 31, 23, 59).atZone(ZoneId.of(timeZone)).toInstant();
72     Instant nextDayAt0000 = LocalDateTime.of(2023, Month.JUNE, 1, 0, 0).atZone(ZoneId.of(timeZone)).toInstant();
73     Instant adjustedTime = TimeHelper.getMidnightAdjustedInstantBasedOnZone(instantAt2359, timeZone, true);
74     assertEquals(nextDayAt0000, adjustedTime);
75 }
76 }

```

## 8. Análise e Resultados

Após a execução dos testes da classe TimeHelperTest.java o seguinte resultado foi obtido na console.

```

> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :compileTestJava UP-TO-DATE
> Task :processTestResources UP-TO-DATE
> Task :testClasses UP-TO-DATE
> Task :componentTests
Jun 19, 2023 7:53:04 PM teammates.common.util.Logger info
INFO: teammates.logic.core.LogicStarter:initializeDependencies:43: Initialized dependencies between logic classes
[=====teammates.common.util.TimeHelperTest=====]
teammates.common.util.TimeHelperTest completed
Jun 19, 2023 7:53:04 PM com.google.cloud.testing.BlockingProcessStreamReader writeLog

```



```
INFO: [datastore] Adding handler(s) to newly registered Channel.
component-tests > component-tests > teammates.common.util.TimeHelperTest > testEndOfYearDates PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest > testFormatDateTimeForDisplay PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest > testGetInstantDaysOffsetBeforeNow PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest > testGetInstantDaysOffsetFromNow PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest > testGetInstantHoursOffsetFromNow PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest > testGetInstantNearestHourBefore PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest > testGetMidnightAdjustedInstantBasedOnZone PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest >
testGetMidnightAdjustedInstantBasedOnZoneWithProjectsTimeConstantShouldReturnSameConstant PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest >
testGetMidnightAdjustedInstantBasedOnZoneWithTime0000notForwardShouldReturnPreviousDayAt2359 PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest >
testGetMidnightAdjustedInstantBasedOnZoneWithTime0059andForwardShouldReturnSameInstant PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest >
testGetMidnightAdjustedInstantBasedOnZoneWithTime0059notForwardShouldReturnSameInstant PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest >
testGetMidnightAdjustedInstantBasedOnZoneWithTime2300andForwardShouldReturnSameInstant PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest >
testGetMidnightAdjustedInstantBasedOnZoneWithTime2300notForwardShouldReturnSameInstant PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest >
testGetMidnightAdjustedInstantBasedOnZoneWithTime2359andForwardShouldReturnNextDayAt0000 PASSED
component-tests > component-tests > teammates.common.util.TimeHelperTest >
testGetMidnightAdjustedInstantBasedOnZoneWithTime2359notForwardFalseShouldReturnSameInstant PASSED
```

Nenhuma falha foi identificada na funcionalidade por meio dos testes realizados, uma vez que todos passaram com sucesso. A criação dos testes aumentou a cobertura de testes do método, conforme ilustrado na figura a seguir.

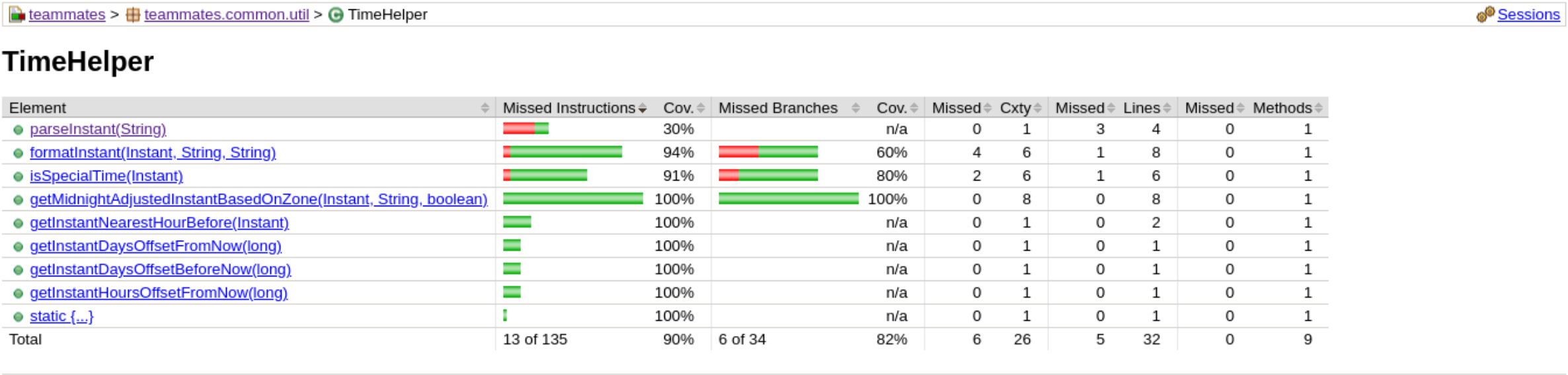


Figura 5. Cobertura de testes depois da adição dos novos testes. (Fonte: elaborado pelo autor, 2023)

```

90.     public static Instant getMidnightAdjustedInstantBasedOnZone(Instant instant, String timeZone, boolean isForward) {
91.         if (isSpecialTime(instant)) {
92.             return instant;
93.         }
94.         ZonedDateTime zonedDateTime = instant.atZone(ZoneId.of(timeZone));
95.         if (isForward && zonedDateTime.getHour() == 23 && zonedDateTime.getMinute() == 59) {
96.             zonedDateTime = zonedDateTime.plusMinutes(1L);
97.         } else if (!isForward && zonedDateTime.getHour() == 0 && zonedDateTime.getMinute() == 0) {
98.             zonedDateTime = zonedDateTime.minusMinutes(1L);
99.         }
100.        return zonedDateTime.toInstant();
101.    }
102.

```

Figura 6. Cobertura de testes depois da adição dos novos testes. (Fonte: elaborado pelo autor, 2023)

## 9. Pull Request

nszchagas

commented in 10 seconds

Description

I increased the test branch coverage in this class from 92% to 100% using the compound condition coverage.

Linked issue

I didn't link any issue because this PR doesn't solve any open issue, just improves the coverage of test.

nszchagas

added 2 commits 15 minutes ago

Improving TimeHelper.getMidnightAdjustedInstantBaseOnZone method test...

994a659

Merge branch 'master' of github.com:nszchagas/teammates

X 4024ea9

TEAMMATES:master

from

nszchagas:master

Open

nszchagas wants to merge 2 commits into TEAMMATES:master from nszchagas:master

Conversation0

Commits2

Checks0

Files changed1

+77-10

Reviewers

No reviews

Still in progress? Convert to draft

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Figura 7. Screenshot do Pull Request. (Fonte: elaborado pelo autor, 2023)

10. Links

Descrição	URL
Fork do projeto	<a href="https://github.com/nszchagas/teammates">https://github.com/nszchagas/teammates</a>
Implementação no projeto	<a href="https://github.com/nszchagas/teammates/blob/master/src/test/java/teammates/common/util/TimeHelperTest.java">https://github.com/nszchagas/teammates/blob/master/src/test/java/teammates/common/util/TimeHelperTest.java</a>
Commit da implementação	<a href="https://github.com/nszchagas/teammates/commit/994a659202bcf2e25adad1450b5a0658eada7510">https://github.com/nszchagas/teammates/commit/994a659202bcf2e25adad1450b5a0658eada7510</a>
Pull Request	<a href="https://github.com/TEAMMATES/teammates/pull/12486">https://github.com/TEAMMATES/teammates/pull/12486</a>