

- pop: Desapila
- 1. Apila a Anacleto
- 2. Comprueba que pop() devuelve a Anacleto
- 3. Apila a Rigoberta y a Florencio.
- 4. Comprueba que pop() devuelve a Florencio
- 5. Comprueba que pop() devuelve a Rigoberta
- 6. Comprueba que pop() devuelve null

```

34 // // Comprueba que top() no devuelve null
35 // // assertEquals(a19camoon.top(), null);
36 // //
37
38 @Test
39 void testPop() {
40     try {
41         // Apila a Anacleto
42         a19camoon.push(personaAnacleto);
43
44         // Comprueba que pop() devuelve a Anacleto
45         assertEquals(a19camoon.pop(), personaAnacleto);
46
47         // Apila a Rigoberta y a Florencio
48         a19camoon.push(personaRigoberta);
49         a19camoon.push(personaFlorencio);
50
51         // Comprueba que pop() devuelve a Florencio
52         assertEquals(a19camoon.pop(), personaFlorencio);
53
54         // Comprueba que pop() devuelve a Rigoberta
55         assertEquals(a19camoon.pop(), personaRigoberta);
56
57         // Comprueba que pop() devuelve null
58         assertEquals(a19camoon.pop(), null);
59     } catch (ElementoNullException e) {
60         // TODO Auto-generated catch block
61         e.printStackTrace();
62     }
63 }
64
65 // @Test
66 // void testPush() {

```

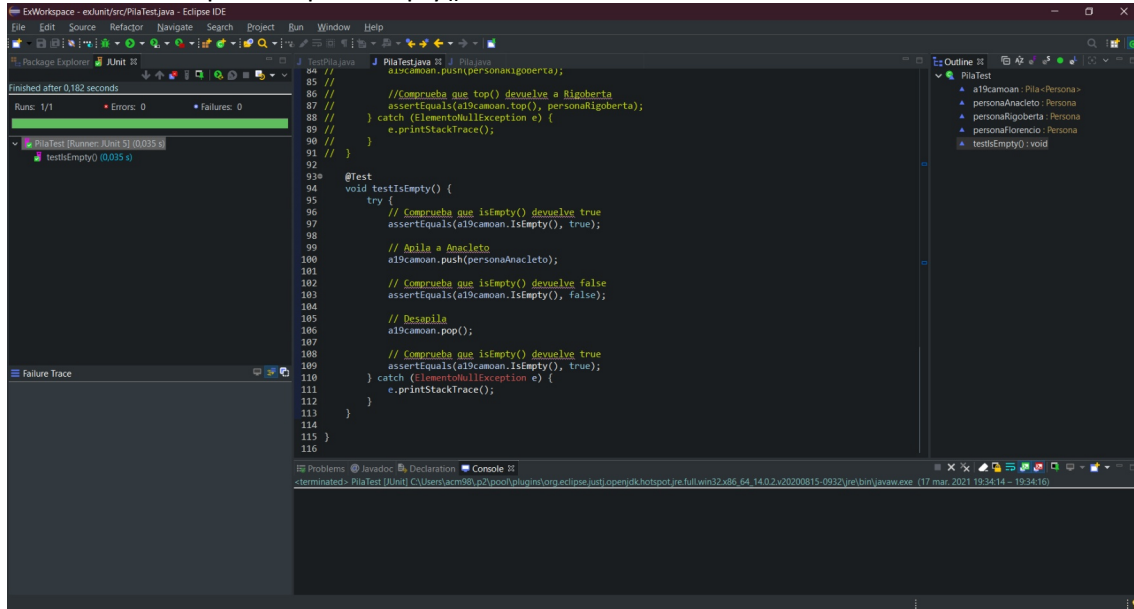
- push: apila
- 1. Comprueba que top() inicialmente devuelve null
- 2. Apila null
- 3. Comprueba que top() devuelve null
- 4. Apila a Anacleto
- 5. Comprueba que top() devuelve a Anacleto
- 6. Apila a Rigoberta
- 7. Comprueba que top() devuelve a Rigoberta

```

61 // // e.printStackTrace();
62 // //
63 // //
64
65 @Test
66 void testPush() {
67     try {
68         // Comprueba que top() inicialmente devuelve null
69         assertEquals(a19camoon.top(), null);
70
71         // Apila null
72         a19camoon.push(null);
73
74         //Comprueba que top() devuelve null
75         assertEquals(a19camoon.top(), null);
76
77         // Apila a Anacleto
78         a19camoon.push(personaAnacleto);
79
80         //Comprueba que top() devuelve a Anacleto
81         assertEquals(a19camoon.top(), personaAnacleto);
82
83         // Apila a Rigoberta
84         a19camoon.push(personaRigoberta);
85
86         //Comprueba que top() devuelve a Rigoberta
87         assertEquals(a19camoon.top(), personaRigoberta);
88     } catch (ElementoNullException e) {
89         e.printStackTrace();
90     }
91 }
92
93 // @Test

```

- isEmpty: comprueba si está vacía
- 1. Comprueba que isEmpty() devuelve true
- 2. Apila a Anacleto
- 3. Comprueba que isEmpty() devuelve false
- 4. Desapila
- 5. Comprueba que isEmpty() devuelve true



Entrega el proyecto, así como los siguientes pantallazos:

1. Cada uno de los métodos implementados
2. La vista con los tiempos en cada uno de los test. Todos han de estar chequeados correctamente, incluida la clase.

