IP2022. Tests de ejecuciones por consola en Jenkins

Estos tests lanzan los programas Java mediante la ejecución del método main de la clase implementada: para ello, automatizan la introducción por teclado de uno o varios valores para cada caso de prueba y comprueban que el valor esperado aparece correctamente como parte de la salida por consola. Los valores utilizados para los casos de prueba son los que aparecen en los guiones de las sesiones, y algunos casos más.

Veamos un ejemplo: para la clase `Euclides.java` del ejercicio 2 de la sesión 3 se ejecuta un caso de prueba en el que se introducen los valores -3, 5 y 25. El resultado debe ser el máximo común divisor de 5 y 25, descartando el -3. En el siguiente ejemplo, el caso de prueba de ha fallado. El mensaje de error tiene la siguiente estructura:

```
Para los valores de entrada -3; 5; 25; el texto esperado no aparece en la salida:
[ESPERADO]
5 y 25 es 5
[SALIDA]
Introduce el primer valor
Introduce el primer valor
Introduce el segundo valor
El maximo comun divisor de 25 y 5 es: 5
```

Así, para cada caso de prueba fallido se muestra:

- los **valores de entrada** del caso de prueba, separados por punto y coma:

```
-3; 5; 25;
```

- El **resultado esperado**, tras el texto [ESPERADO]

```
5 y 25 es 5
```

- La **salida por consola** de la ejecución del programa, tras el texto [SALIDA]

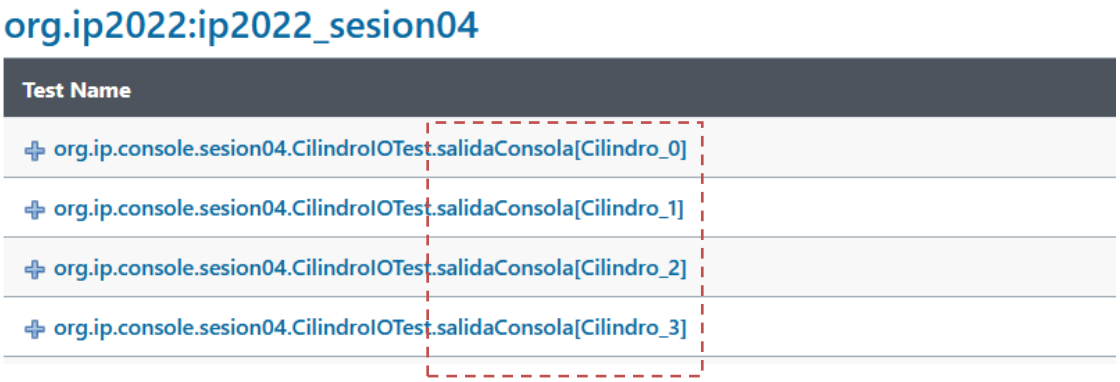
```
Introduce el primer valor
Introduce el primer valor
Introduce el segundo valor
El maximo comun divisor de 25 y 5 es: 5
```

El motivo del fallo es que el texto ESPERADO debe aparecer como parte del texto de salida, y no es así. En este ejemplo, en la salida se ha añadido el carácter ':' (dos puntos) que no está en el texto esperado, por lo que la prueba ha fallado. La salida debe ser exactamente igual a la que se muestra en los enunciados de las sesiones.

En estos test de consola (consoleIO), para tener acceso a la entrada por teclado y la salida por consola en JUnit utilizamos el framework System Rules (https://stefanbirkner.github.io/system-rules/).

Además, para ejecutar el mismo test con distintos casos de prueba (valores de entrada y resultado esperado) utilizamos tests parametrizados de JUnit (https://www.tutorialspoint.com/junit/junit_parameterized_test.htm).

Estos tests por consola tienen un nombre y un número como sufijo para cada caso de prueba:



Se han incorporado tests de consola a partir de la **sesión 03** en adelante. A continuación, os exponemos algunos ejemplos de fallos que se producen en este tipo de tests. Veréis que algunos fallos se pueden considerar *falsos positivos*, es decir, el test falla cuando no debería fallar. Aunque hemos tratado de minimizarlos al máximo, **en algunas ocasiones el mensaje de salida del ejercicio implementado por el estudiante no coincide con la salida esperada según el enunciado**, así que el **test falla**, como en el ejemplo anterior.

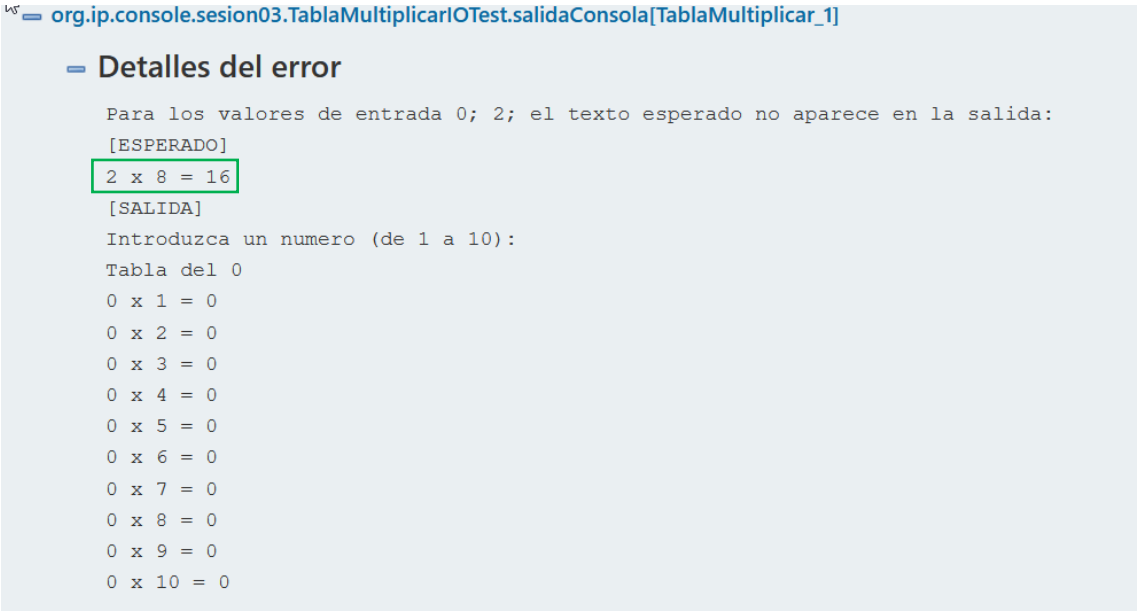
Insistimos en que la salida debe ser **exactamente igual** a la que se muestra en los enunciados de las sesiones.

A continuación, se muestran ejemplos de fallos habituales detectados en las sesiones 03 y 04.

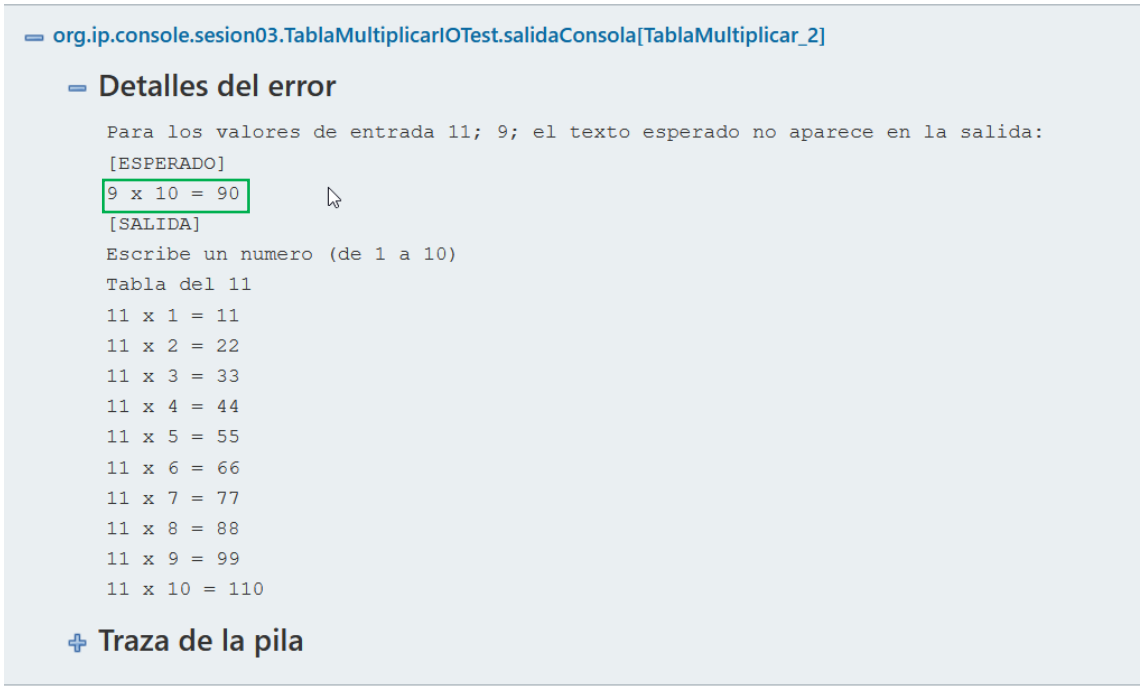
Sesion 03: TablaMultiplicar.java

Falla cuando no se controla que el valor de entrada esté entre 1 y 10.

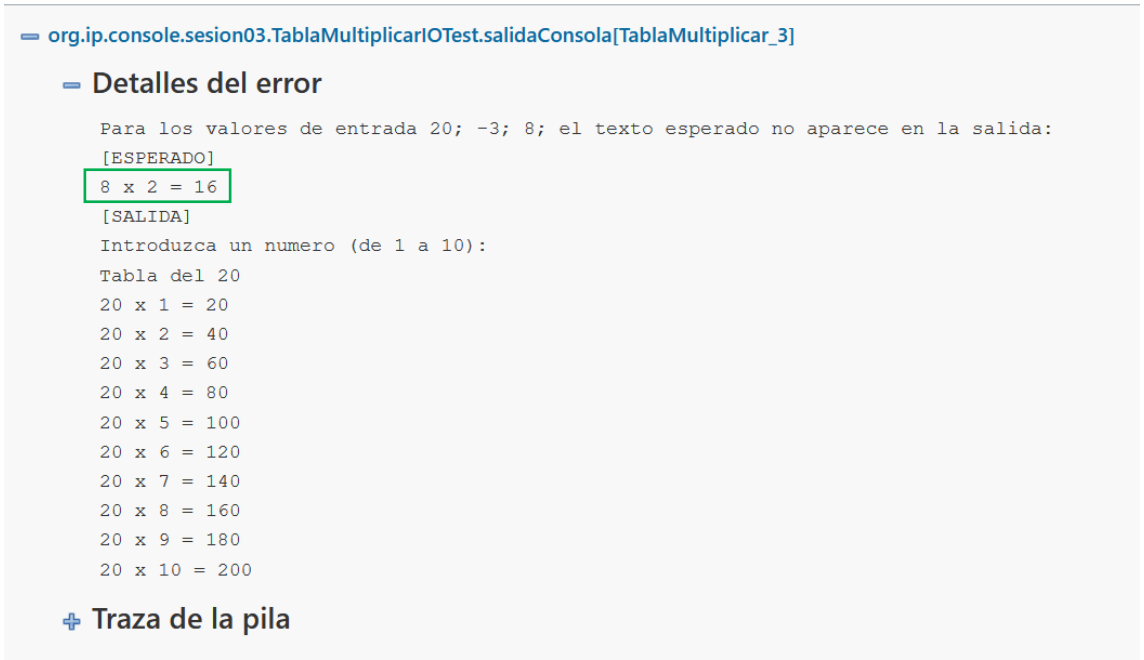
- **Ejemplo 1:** El caso de prueba 1 introduce el valor 0, que debería ser controlado.



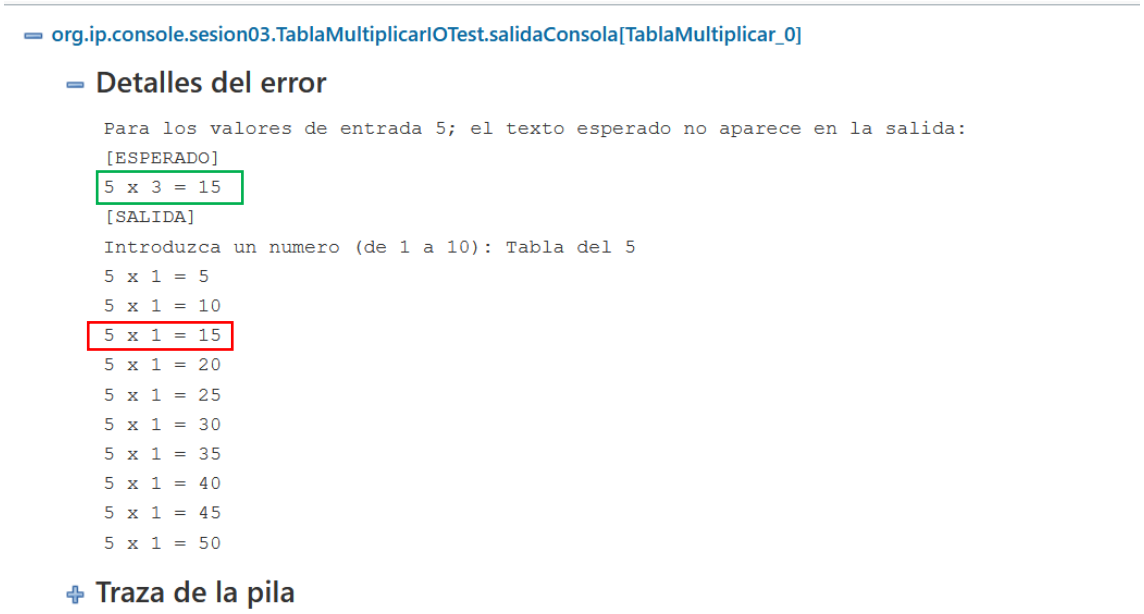
- **Ejemplo 2:** El caso de prueba 2 introduce el valor 11, que debería ser controlado.



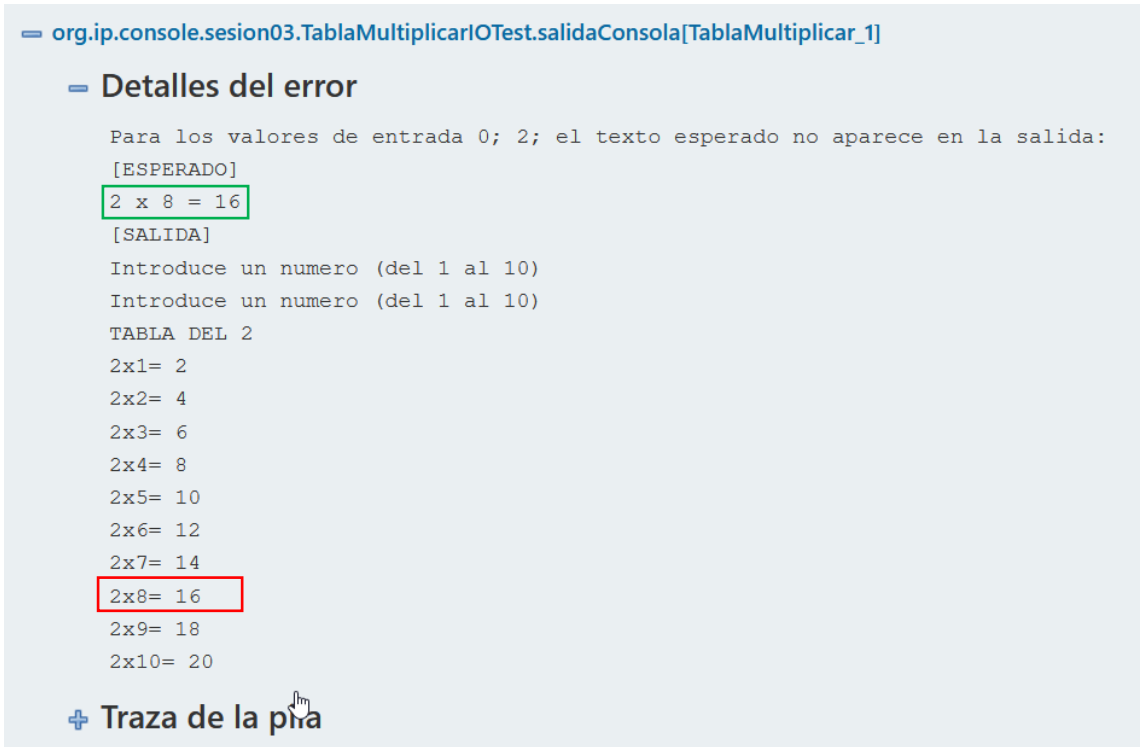
- **Ejemplo 3:** El caso de prueba 3 introduce los valores 20 y -3 que deberían ser controlados:



Ejemplo 4: Este test también falla cuando la salida por pantalla no es correcta:

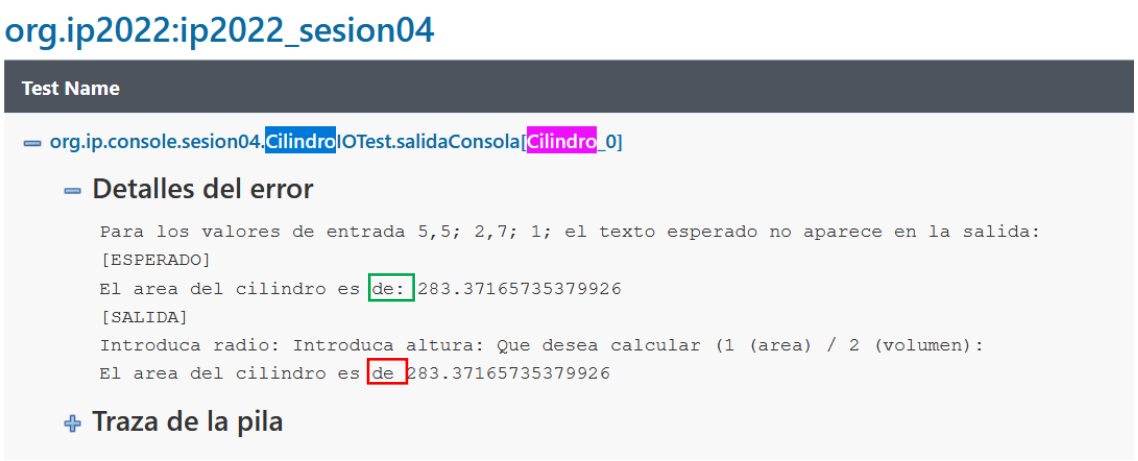


Ejemplo 5: El test también falla si no se dejan los espacios en blanco adecuados. Insistimos, los mensajes de salida deben ser los mismos (**exactamente iguales**) que se proporcionan en los guiones de prácticas.

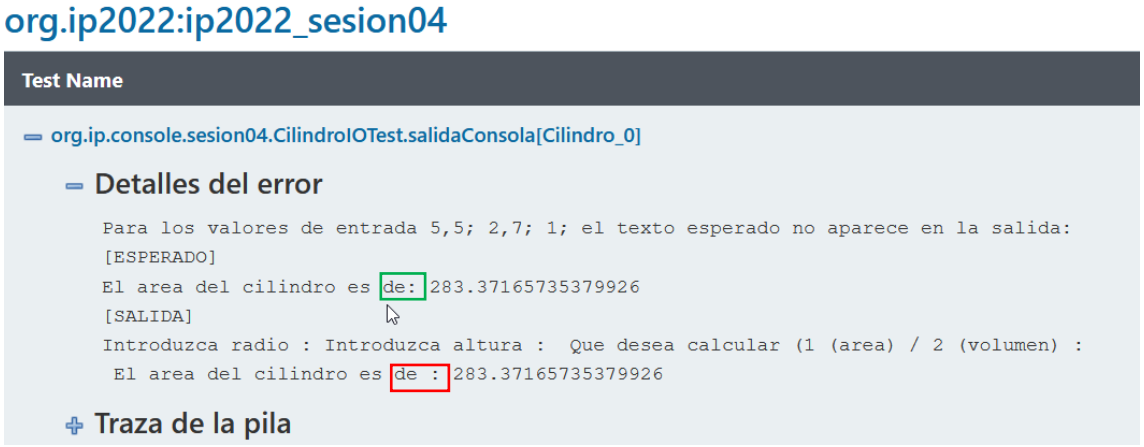


Sesion 04: Cilindro.java

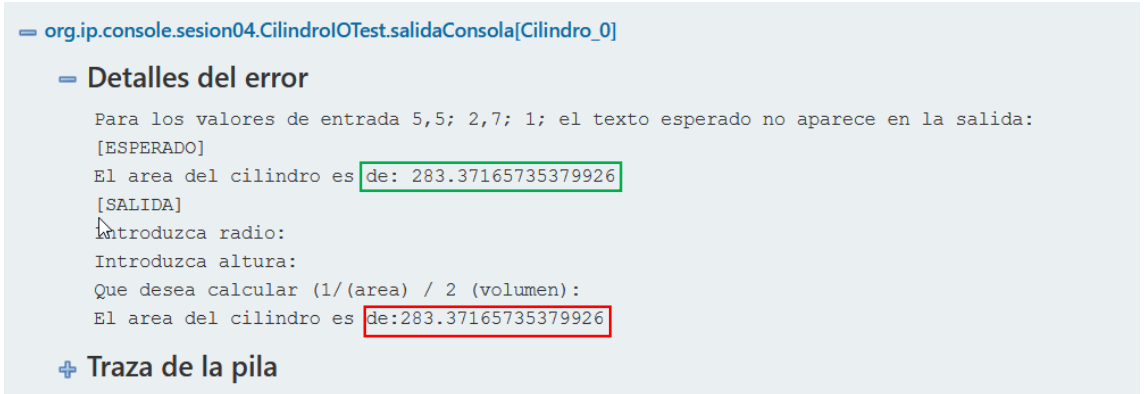
Ejemplo 1: El test falla si el estudiante ha olvidado poner dos puntos (:) en la salida.



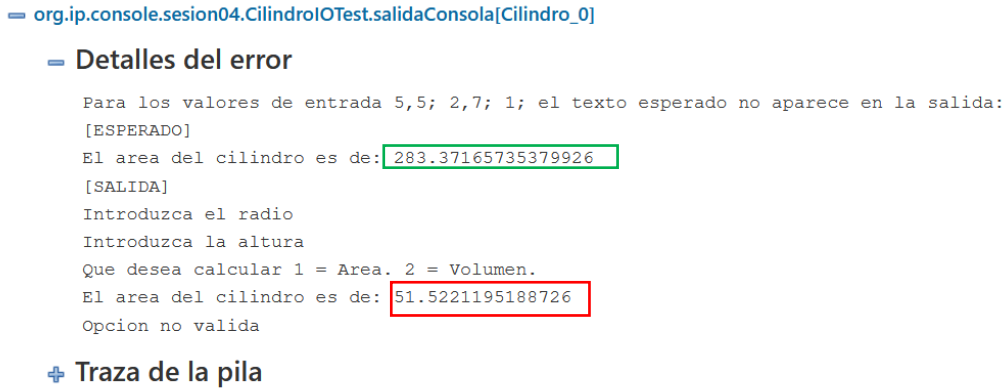
Ejemplo 2: El test falla si el estudiante añade espacio en blanco antes de los dos puntos.



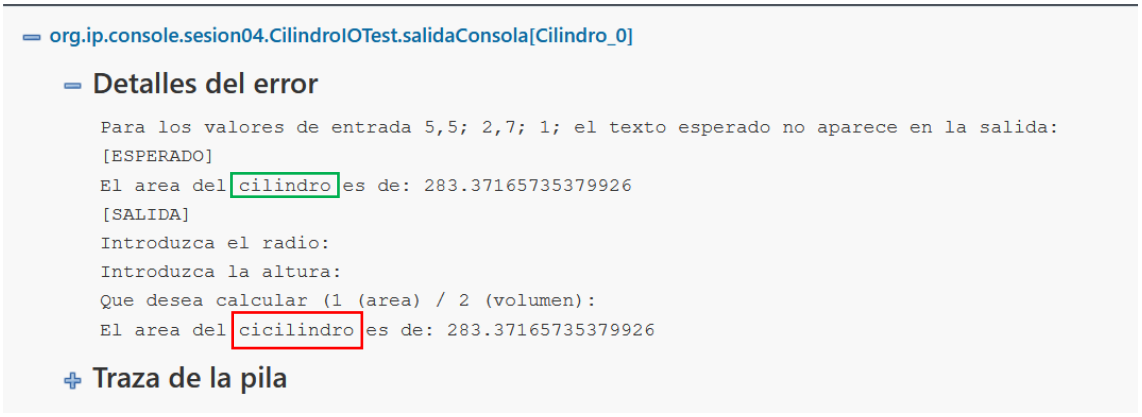
Ejemplo 3: El test falla si el estudiante no añade espacio en blanco después de los dos puntos.



Ejemplo 4: El test falla si el resultado de los cálculos a realizar no es el correcto:

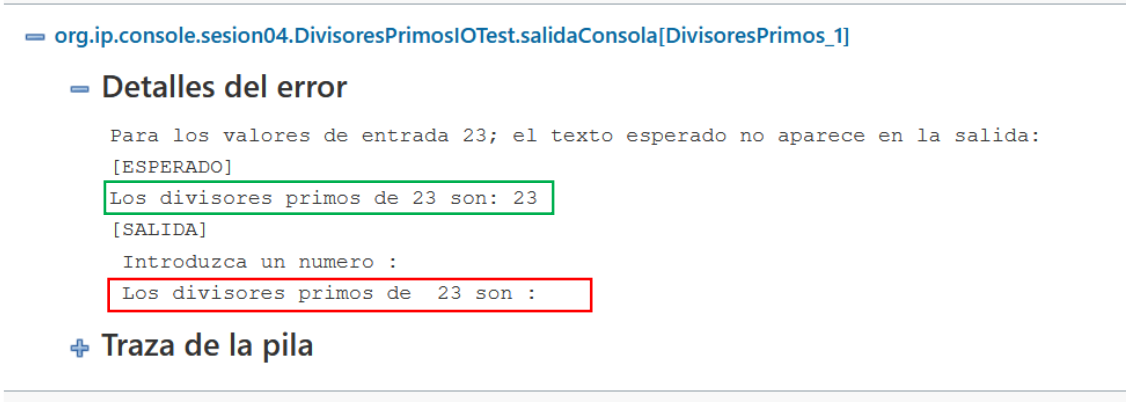


Ejemplo 5: El test falla si se comente errores ortográficos.

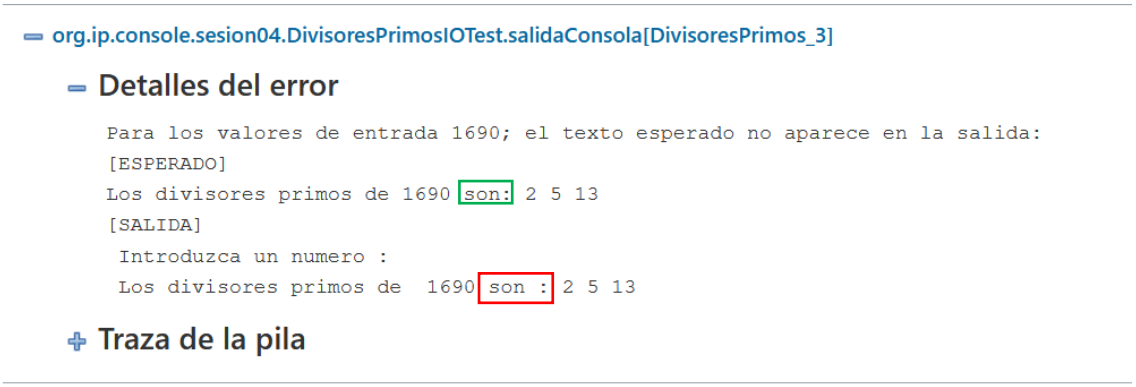


Sesion 04: DivisoresPrimos.java

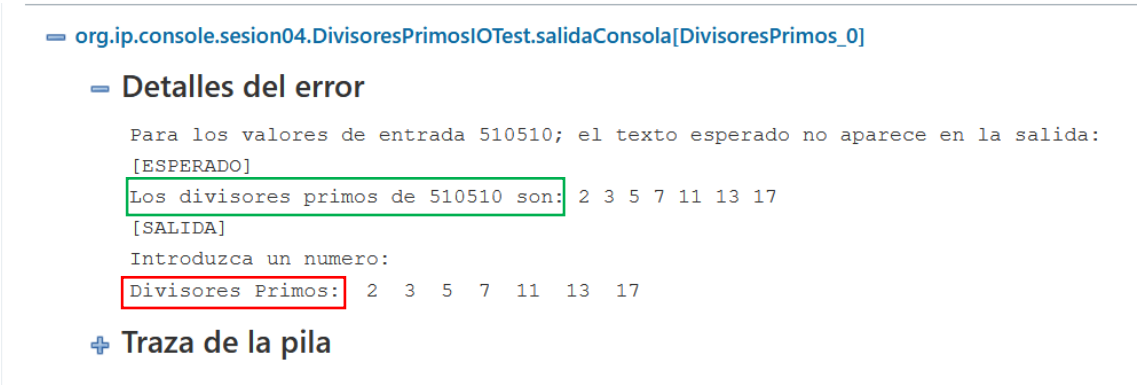
Ejemplo 1: El test falla si el resultado no es correcto.



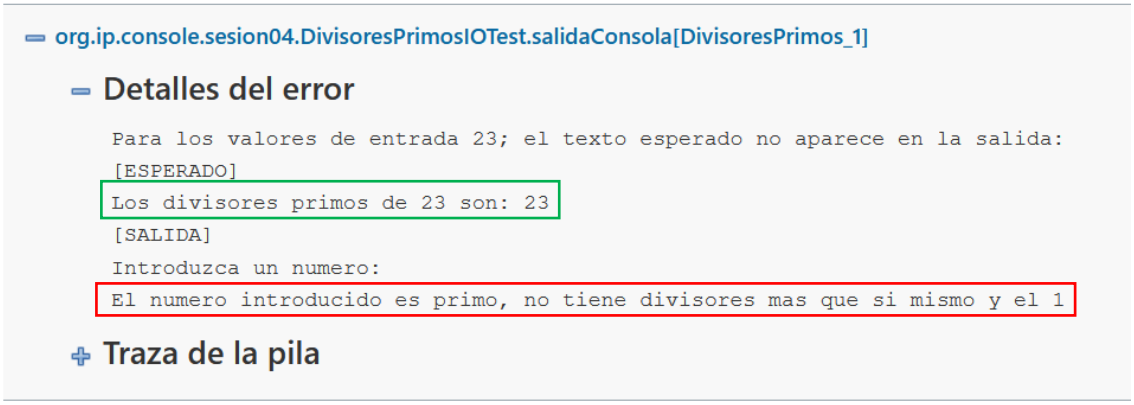
Ejemplo 2: El test falla si el estudiante añade espacio en blanco entre la palabra “son” y los dos puntos.



Ejemplo 3: El test falla si se faltan palabras en la salida. Insistimos, los mensajes de salida deben ser los mismos (**exactamente iguales**) que se proporcionan en los guiones de prácticas.



Ejemplo 4: El test falla si en la salida se muestra un contenido que no es el esperado, aunque signifique lo mismo. Insistimos, los mensajes de salida deben ser los mismos (**exactamente iguales**) que se proporcionan en los guiones de prácticas.



Ejemplo 5: El test falla si se cometen errores ortográficos. Insistimos, los mensajes de salida deben ser los mismos (**exactamente iguales**) que se proporcionan en los guiones de prácticas.