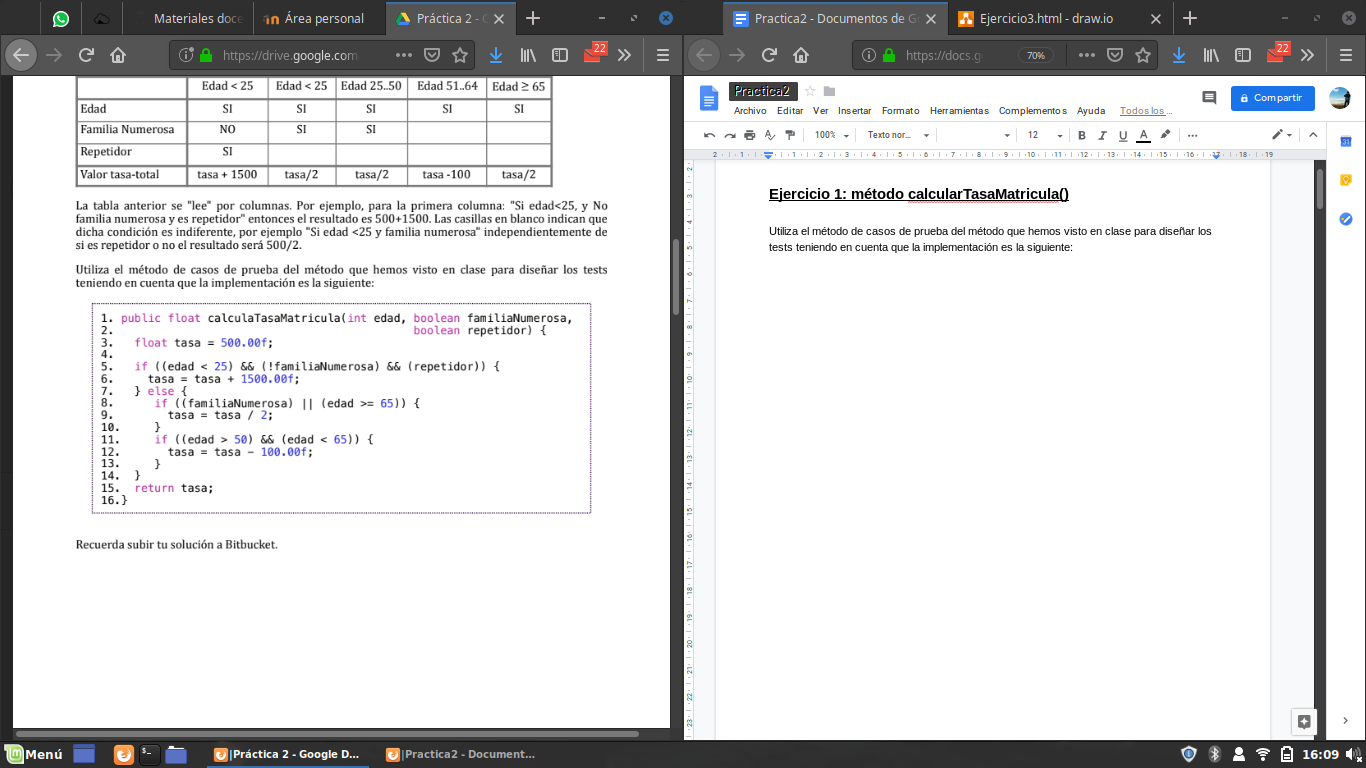
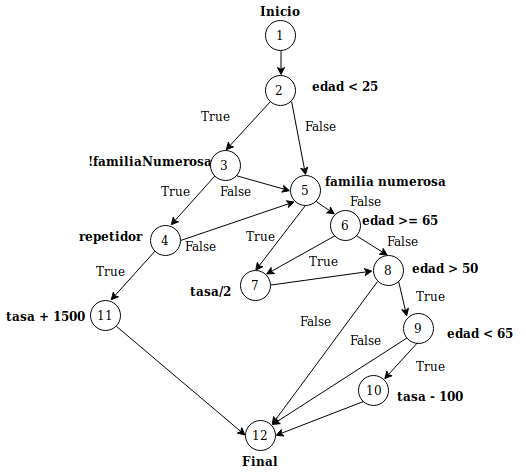
1. **P02-Diseño de pruebas de caja blanca**
   1. **Ejercicio 1: método calcularTasaMatricula()**

**Queremos diseñar los casos de prueba para el método calculaTasaMatricula() (la especificación es la misma que la del ejercicio 3 de la práctica anterior).**

**Utiliza el método de casos de prueba del método que hemos visto en clase para diseñar los tests teniendo en cuenta que la implementación es la siguiente:**



El grafo es el siguiente:



CC = 18-12+2 -> 8

C1 = 1 - 2 - 3 - 4 - 11

C2 = 1 - 2 - 3 - 5 - 7 - 12

C3 = 1 - 2 - 3 - 4 - 5 - 6 - 8 - 12

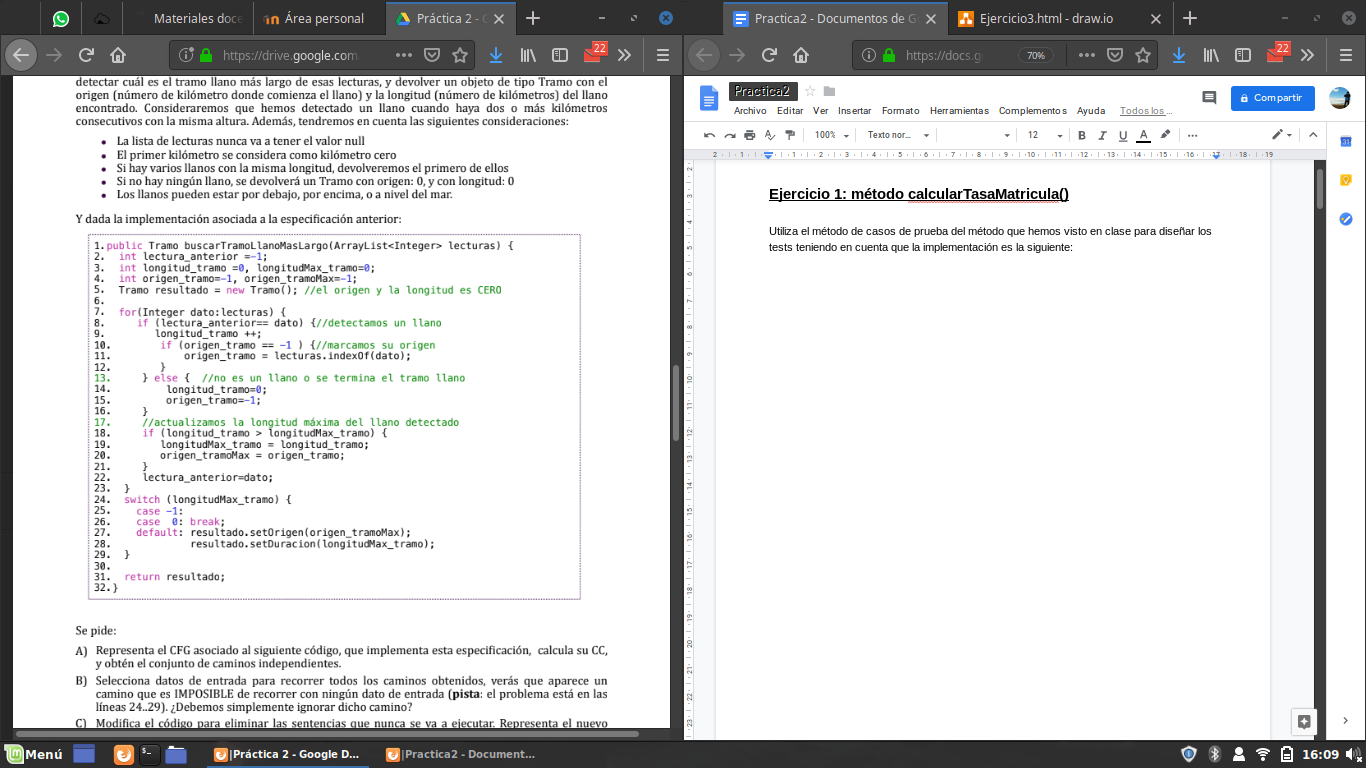
C4 = 1 - 2 - 5 - 6 - 7 - 8 - 9 - 10 - 12

C5 = 1 - 2 - 5 - 6 - 7 - 8 - 9 - 12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identificador del Caso de prueba** | **Datos de entrada** | | | **Resultado esperado** |
| **Edad** | **Familia numerosa** | **Repetidor** |
| **C1** | 24 | false | true | 2000 |
| **C2** | 24 | true | false | 250 |
| **C3** | 24 | false | false | 500? |
| **C4** | 55 | true | false | 250 |
| **C5** | 66 | true | false | 400 |

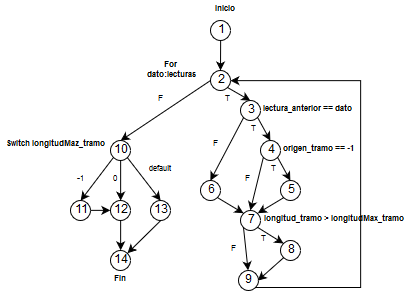
**Ejercicio 2: método buscarTramoLlanoMasLargo()**

* 1. **Se proporciona una especificación para el método buscarTramoLlanoMasLargo() al igual que el siguiente código:**



**Se pide:**

**A) Representa el CFG asociado al siguiente código, que implementa esta especificación, calcula su CC, y obtén el conjunto de caminos independientes.**

****

CC = 19 - 14 + 2 -> 7

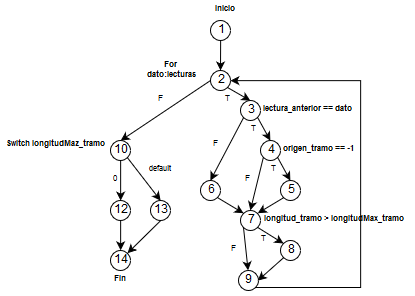
C1 = 1 - 2 - 3 - 6 - 7 - 9 - 2 - 10 - 12 - 14 {1}

C2 = 1 - 2 - 3 - 6 - 7 - 9 - 2 - 3 - 4 - 5 - 7 - 8 - 9 - 2 - 3 - 4 - 7 - 8 - 9 - 2 - 10 - 13 - 14 {3,3,3}

**B) Selecciona datos de entrada para recorrer todos los caminos obtenidos, verás que aparece un camino que es IMPOSIBLE de recorrer con ningún dato de entrada (pista: el problema está en las líneas 24..29). ¿Debemos simplemente ignorar dicho camino?**

Deberíamos ignorar dicho camino ya que no va a pasar nunca. LongitudMax\_tramo no va a decrementar en ninguna parte del código.

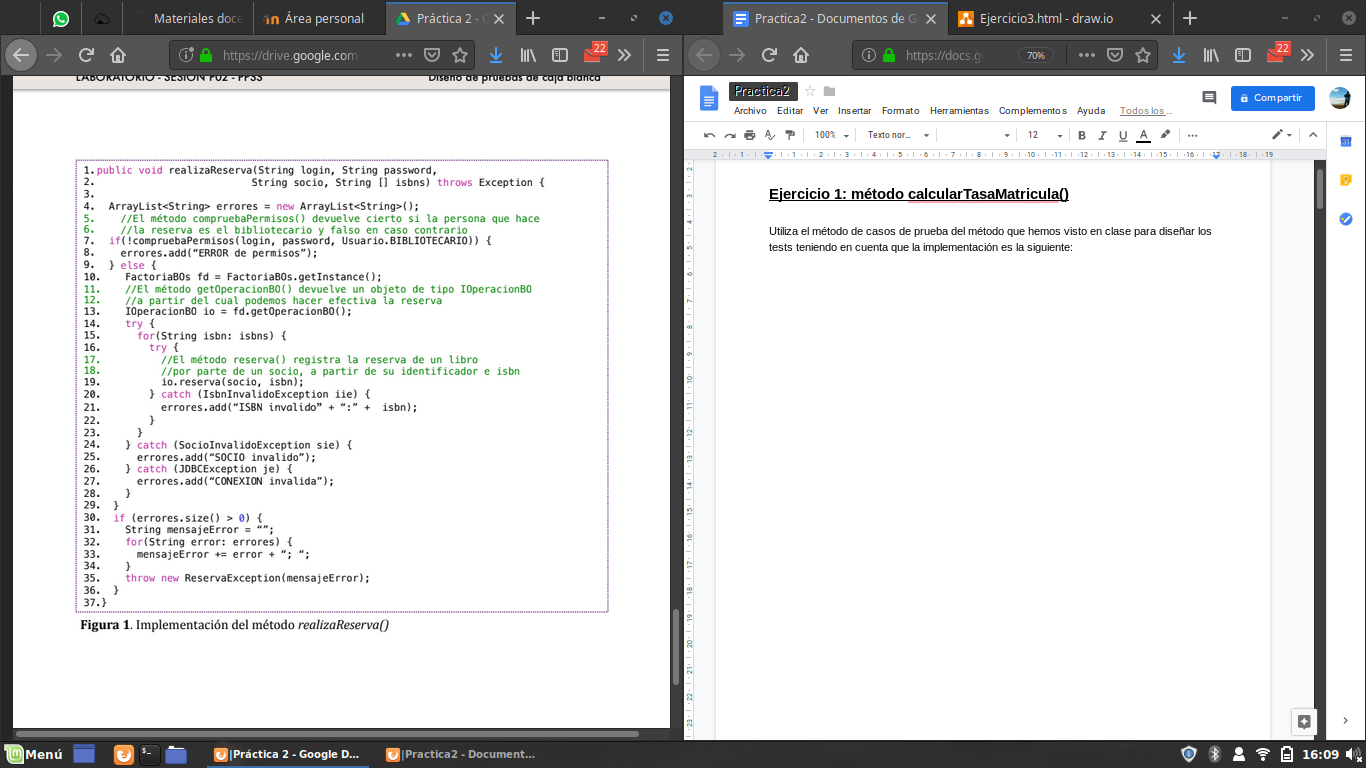
**C) Modifica el código para eliminar las sentencias que nunca se va a ejecutar. Representa el nuevo grafo y diseña de nuevo los casos de prueba. ¿Podemos asegurar que el código es correcto? Piensa en algún caso de prueba adicional que pudiera poner de manifiesto un defecto en nuestro código.**

****

|  |  |  |
| --- | --- | --- |
| **Identificador del Caso de prueba** | **Datos de entrada** | **Resultado esperado** |
| **Array Lectura** |
| **C1** | {1} | 0 - 0 |
| **C2** | {3,3,3} | 0 - 3 |

* 1. **Ejercicio 3: método realizarLizaReserva()**

Se proporciona una especificación para el método realizaReserva() y además el siguiente código:



A partir del código y la especificación proporcionadas, diseña una tabla de casos de prueba para el método realizaReserva().

CC = 20 - 14 + 2 -> 8

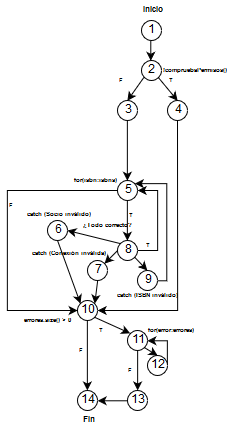
C1 = 1 - 2 - 3 - 5 - 8 - 9 - 5 - 10 - 14 (Todos los datos correctos)

C2 = 1 - 2 - 4 - 10 - 11 - 12 - 11 - 13 - 14 (No sea el bibliotecario)

C3 = 1 - 2 - 3 - 5 - 8 - 6 - 10 - 11 - 12 - 13 -14 (Socio inválido)

C4 = 1 - 2 - 3 - 5 - 8 - 9 - 5 - 10 - 11 - 12 - 11 - 13 - 14 (Isbn Inválido)

C5 = 1 - 2 - 3 - 5 - 8 - 7 - 10 - 11 - 12 - 11 - 13 - 14 (Conexión inválida)



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identificador del Caso de prueba** | **Datos de entrada** | | | | **Resultado esperado** |
| **Login** | **Password** | **Socio** | **ISBN** |
| **C1** | biblio | biblio | socio | {12345} | (Nada) |
| **C2** | \*nbiblio | \*nbiblio | socio | {12345} | “ERROR de permisos” |
| **C3** | biblio | biblio | \*nsocio | {12345} | “SOCIO inválido” |
| **C4** | biblio | biblio | socio | \*{23456} | “ISBN inválido: 23456” |
| **C5** | biblio | biblio | socio | {12345} | “CONEXIÓN inválida” |

\*(Suponemos que el ISBN “23456” es incorrecto)

\*(Suponemos que el login y password “nbiblio” es incorrecto)

\*(Suponemos que el socio“nsocio” es incorrecto)

\*(Todo lo demás es correcto)