Prueba Unitaria de caja Blanca.

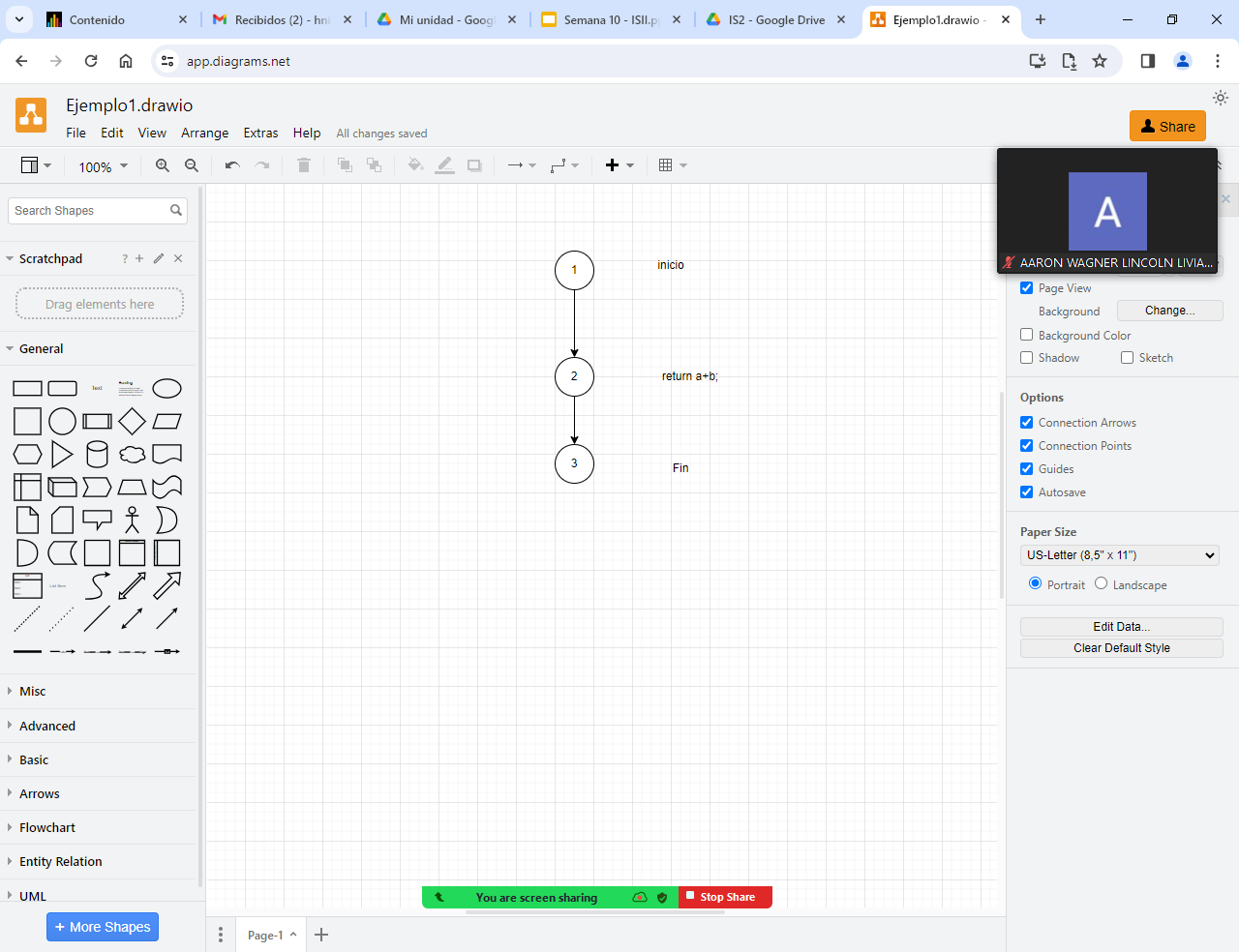
Codigo:

1. public static int suma(int a, int b){

2. return a+b;

3. }

Paso 1: Crear un grafo de flujo



Paso 2. Hallar la Complejidad Ciclomática

V(G) = 2 – 3 + 2 = 1

Cuantos Caminos independiente puedo identificar en el Grafo.

Paso 3. Identificar Caminos Independientes

Camino 1: **1 – 2 – 3**.

Paso 4. Identificar Casos de prueba.

Caso de Prueba:

Entradas

A = -4;

B= 7;

Salidas:

R = 3

Ejercicio 2:

**Ejercicio 2: Función con Condicionales**

1.    public String determinarSigno(int numero) {

2.       if (numero > 0) {

3.            return "Positivo";

4.       } else if (numero < 0) {

5.            return "Negativo";

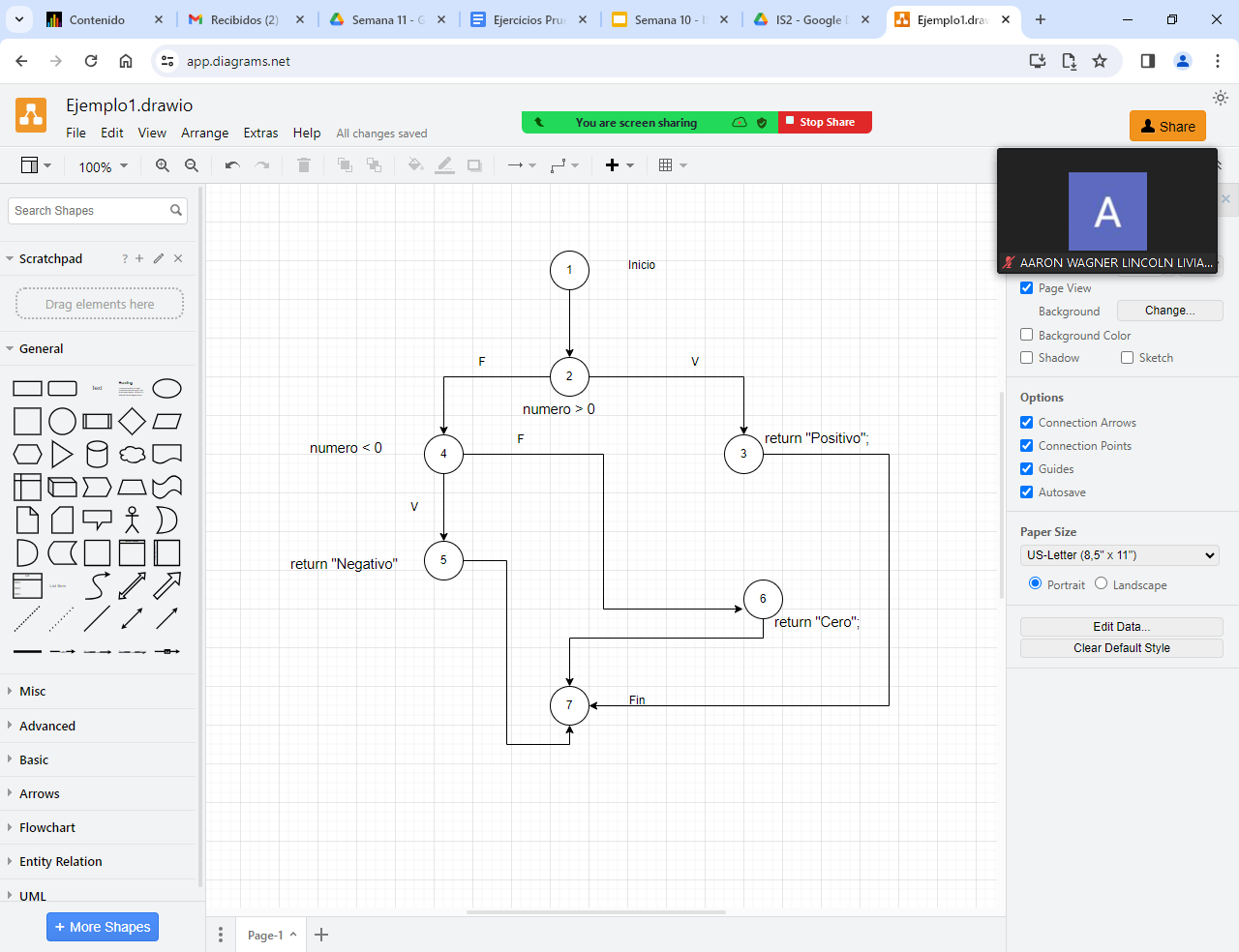
6.       } else {

              return "Cero";

         }

7.    }

Paso 1. Crear el grafo de Flujo



Paso 2:

V(G) = 8 – 7 + 2 = 3

Paso 3 y 4: Identificar caminos independiente y casos de prueba

|  |  |  |  |
| --- | --- | --- | --- |
| Nro. | Camino | Entradas | Salidas |
| 1 | 1 – 2 – 3 – 7 | Numero = 7 | “Positivo” |
| 2 | 1 -2 – 4 – 5 -7 | Numero = -5 | “Negativo” |
| 3 | 1 – 2 – 4 – 6 – 7 | Numero = 0 | “Cero” |

Ejercicio 3: Condicionales con predicados

**(1.)**    public boolean funcionConPredicados(boolean condicion1, boolean condicion2) {

        if (**(2.)**condicion1 && **(3.)**condicion2) {

4.            return true;

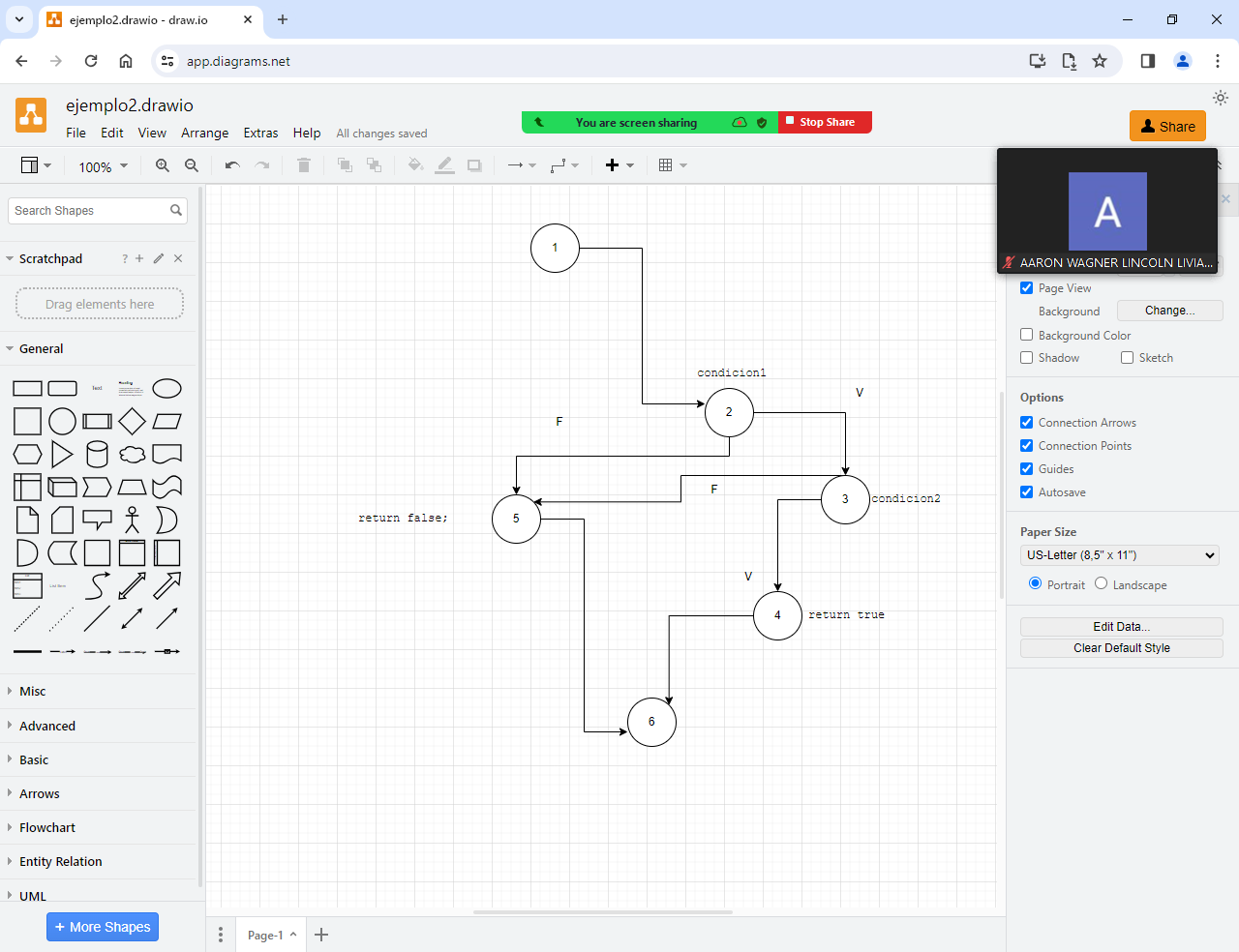
5.        } else {

            return false;

        }

6.    }

Paso 1: Crear el Grafo de Flujo.



Paso 2: Complejidad Ciclomatica:

V(G) = 7 – 6 + 2 = 3

Paso 3 y 4: Identificar caminos independiente y casos de prueba

|  |  |  |  |
| --- | --- | --- | --- |
| Nro. | Camino | Entradas | Salidas |
| 1 | 1 – 2 – 3 -4 – 6 | Con1 = true y Con2 = true | True |
| 2 | 1 – 2 – 5 – 6 | Con1 = false y Con2 = true | False |
| 3 | 1 -2 – 3 – 5 - 6 | Con1 = true y Con2 = false | False |

Ejercicio 4:  
1    public int calcularFactorial(int numero) {

2        if (numero < 0) {

3            throw new IllegalArgumentException("El número debe ser no negativo.");

        }

4        int factorial = 1;

       for ((5) int i = 1; (6) i <= numero; (8) i++) {

            (7) factorial \*= i;

        }

        (9) return factorial;

10.   }

Paso 2:

V(G) = 11 – 10 + 2 = 3

Paso 3 y 4: Identificar caminos independiente y casos de prueba

|  |  |  |  |
| --- | --- | --- | --- |
| Nro. | Camino | Entradas | Salidas |
| 1 | 1 – 2 – 3 -10 | Numero = -5 | Error |
| 2 | 1 – 2 – 4 – 5 – 6 – 9 – 10 | Numero = 0 | 1 |
| 3 | 1 – 2- 4 -5 **-6 – 7 – 8** – 6 – 9 - 10 | Numero = 2 | 2 |