**UNIVERSIDAD DE LAS FUERZAS ARMADAS**

**ESPE**

****

**PRUEBA DE CAJA BLANCA**

**“Aplicativo de gestión y administración de información de MUNDOTV-EC”**

**VERSIÓN 2**

**ASIGNATURA:**

Metodología de Desarrollo de Software

**ESTUDIANTES:**

Carlos Alejandro Pullas Herrera

Michael Alexander Villacrés Pabón

**NRC 9899**

**Ing. Jenny A. Ruiz**

**Sangolquí- Ecuador, 21 de junio de 2023**

**Login de ingreso el aplicativo**

**Caso de Prueba 1**

**CÓDIGO FUENTE:**

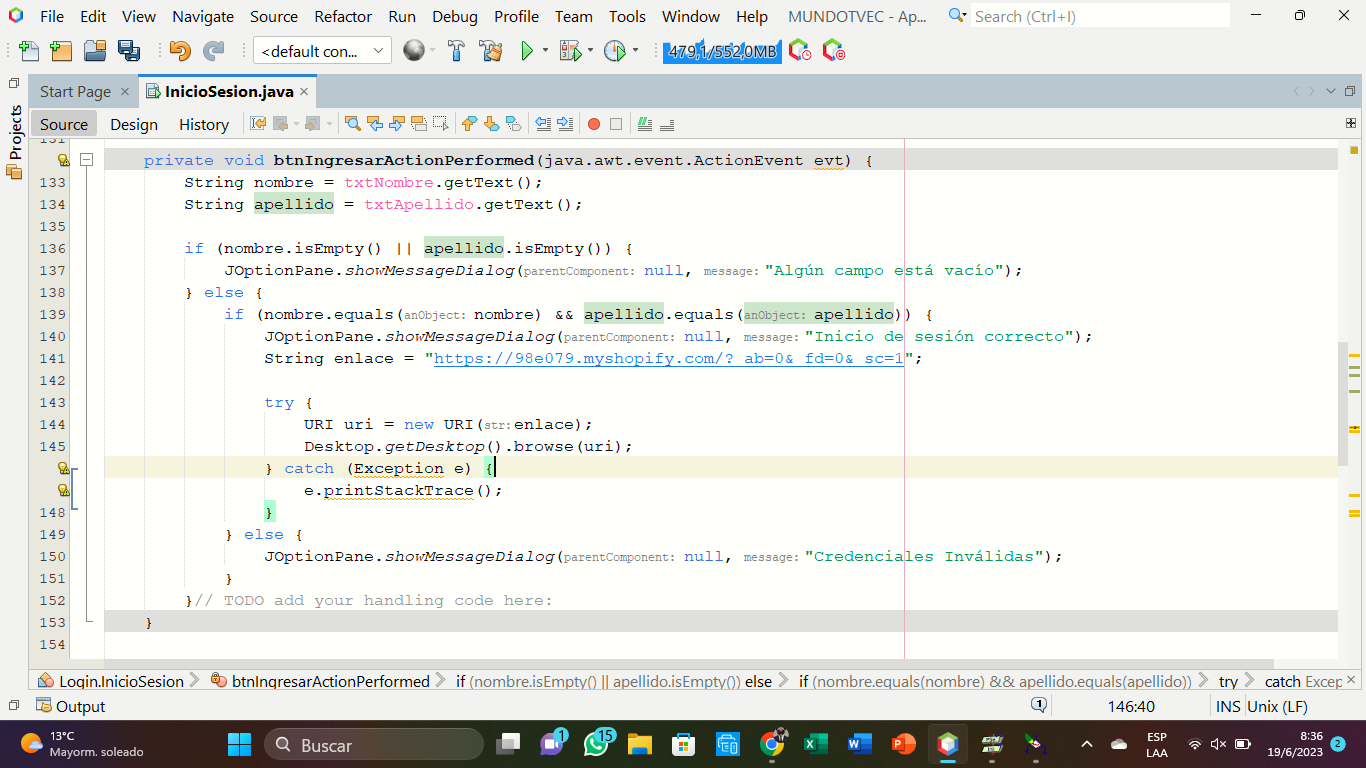


Ilustración 1. Código Login ingreso del aplicativo

**DIAGRAMA DE FLUJO:**

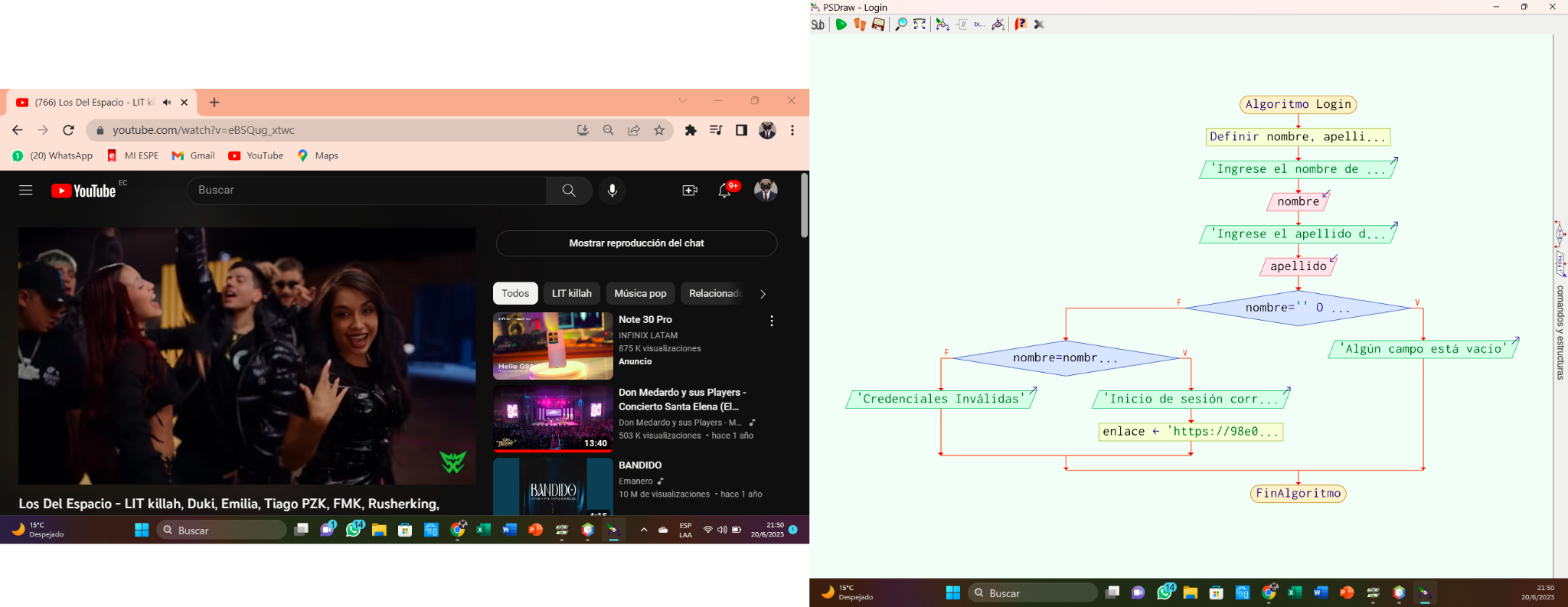


Ilustración 2. Diagrama de flujo login

**DIAGRAMA DE GRAFOS:**

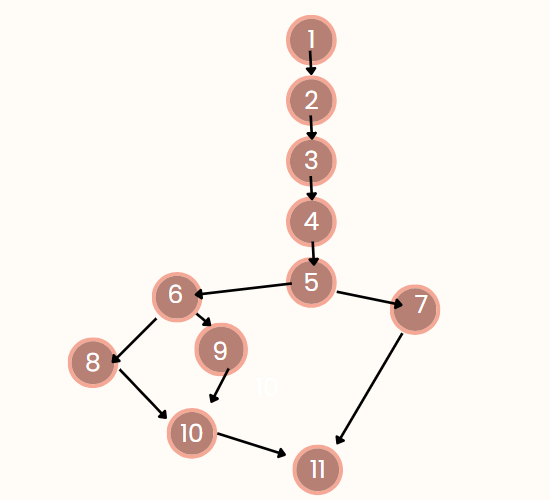


Ilustración 3. Diagrama de Grafos login

**RUTAS**

**R1:** 1,2,3,4,5,7,11

**R2:** 1,2,3,4,5,6,9,10,11

**R3:** 1,2,3,4,5,6,8,10,11

**COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

●   V(G) = número de nodos predicados(decisiones) 2 +1 = 3

●   V(G) = E – N + 2 = 12 – 11 + 2 = 3

**DONDE:**

**P:** Número de nodos predicadoS

**E:** Número de aristas

**N:** Número de nodos

**P:** 2

**E:** 12

**N:** 11

**CONCLUSIONES:**

* Al aplicar la técnica de prueba de software que se centra en examinar y evaluar la estructura interna del código fuente de un programa junto con la complejidad ciclomática se ha determinado la cantidad y la extensión de las pruebas necesarias para garantizar una cobertura adecuada de todos los caminos de ejecución posibles que en este caso nos genera 3 rutas las cuales son las siguientes R1: 1,2,3,4,5,7,11; R2: 1,2,3,4,5,6,9,10,11; R3: 1,2,3,4,5,6,8,10,11
* La realización de pruebas de caja blanca no solo implica la evaluación del comportamiento del usuario en la interfaz, sino también la detección de errores en el código fuente. No es factible asegurar que un software o sistema nunca falle, solamente se pueden llevar a cabo pruebas que reduzcan dicho riesgo.