**Informe Pruebas de Software**

**Registro Ágil**

Alejandra Parias Botero

Valentina Herrera Calle

Juliana Robledo Bedoya

Servicio Nacional de Aprendizaje.

Centro Textil y de Gestión Industrial

Análisis y Desarrollo de Sistemas de Información

Medellín

2020

**Informe Pruebas de Software**

**Registro Ágil**

Alejandra Parias Botero

Valentina Herrera Calle

Juliana Robledo Bedoya

Instructor(es):

Lee Jared Escobar Gómez

Alejandro Mejía Jaramillo

Edilfredo Pineda Flores

Aharon Alexander Aguas

Servicio Nacional de Aprendizaje.

Centro Textil y de Gestión Industrial

Análisis y Desarrollo de Sistemas de Información

Medellín

2020.

**Tabla de contenido**

[**Introducción** 5](#_Toc42020739)

[**Objetivo general** 6](#_Toc42020740)

[**Objetivos específicos** 6](#_Toc42020741)

[**Pruebas de caja blanca** 7](#_Toc42020742)

[**Modulo usuario** 7](#_Toc42020743)

[Registrar elemento 7](#_Toc42020744)

[Editar elemento 9](#_Toc42020745)

[**Modulo vigilante** 9](#_Toc42020746)

[Ingreso - salida elementos 9](#_Toc42020747)

[Historial 10](#_Toc42020748)

[**Modulo supervisor** 10](#_Toc42020749)

[Registrar vigilante 10](#_Toc42020750)

[Activar – desactivar vigilante 11](#_Toc42020751)

[**Pruebas de caja negra** 12](#_Toc42020752)

[**Modulo usurario** 12](#_Toc42020753)

[Registrar – editar elemento 12](#_Toc42020754)

[**Modulo vigilante** 14](#_Toc42020755)

[Verificación 14](#_Toc42020756)

[**Modulo supervisor** 15](#_Toc42020757)

[Registrar y listar vigilante 15](#_Toc42020758)

[**Casos de prueba** 18](#_Toc42020759)

[**Modulo usuario** 18](#_Toc42020760)

[Registrar elemento 18](#_Toc42020761)

[**Modulo vigilante** 20](#_Toc42020762)

[Verificación 20](#_Toc42020763)

[**Modulo supervisor** 21](#_Toc42020764)

[Registrar – listar vigilante 21](#_Toc42020765)

[**Conclusión** 23](#_Toc42020766)

# **Introducción**

El presente documento brindara información sobre las pruebas de software caja blanca, caja negra y casos de prueba desarrollados al sistema ‘Registro Ágil’, mostrando una descripción detallada del procedimiento que se llevó a cabo para realizar cada prueba. Se mostrara algunos fragmentos de pseudocódigo, datos de entrada y salida, requisitos para realizar algunos procedimientos y que puede ocurrir si no se cumplen dichos requisitos. Realizando estas pruebas a los módulos más importantes de la plataforma.

# **Objetivo general**

Verificar si las funcionalidades de los diferentes módulos de la plataforma se están ejecutando correctamente y que posibles falencias se pueden encontrar durante la funcionalidad de la misma, probando su calidad y fiabilidad.

**Objetivos específicos**

* Realizar un paso a paso de las principales funcionalidades del sistema, llevando dicho proceso al extremo y ver cómo reacciona a los diferentes datos de entrada o caminos.
* Mostrar y analizar los errores que se pueden encontrar en el sistema y no se habían descubierto hasta entonces.
* Verificar que todos los requisitos y/o requerimientos del sistema se estén implementando correctamente.

# **Pruebas de caja blanca**

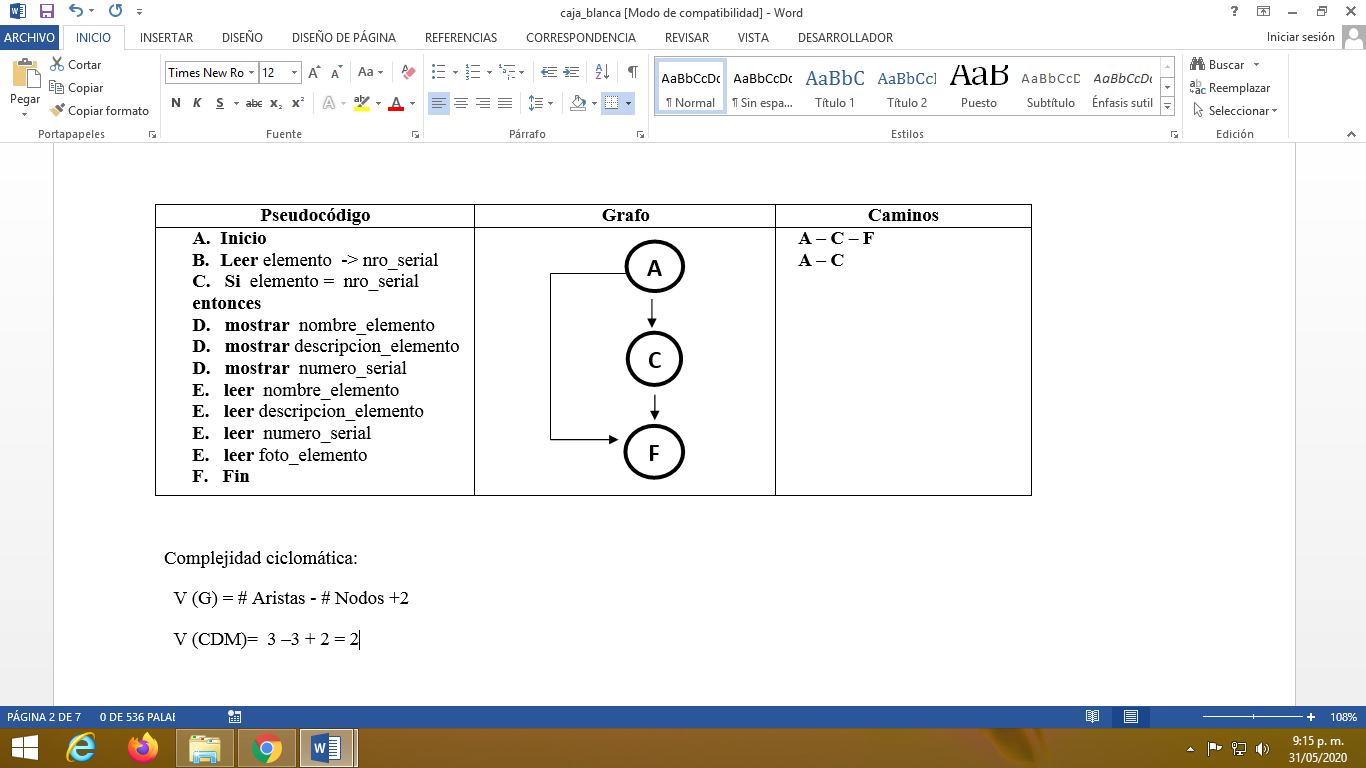
Las pruebas de caja blanca se centran en los procedimientos lógicos del sistema. Para realizar las pruebas del sistema ‘registro Ágil nos enfocamos en la técnica del camino básico, en la que se debe analizar y examinar todos los caminos lógicos posibles y realizar los diferentes recorridos de un fragmento de código, garantizando que cada sentencia se recorre al menos una vez.

Para su correcta elaboración primero se realiza el grafo de flujo, luego se calcula la complejidad ciclomática y se determina un conjunto de caminos.

## **Modulo usuario**

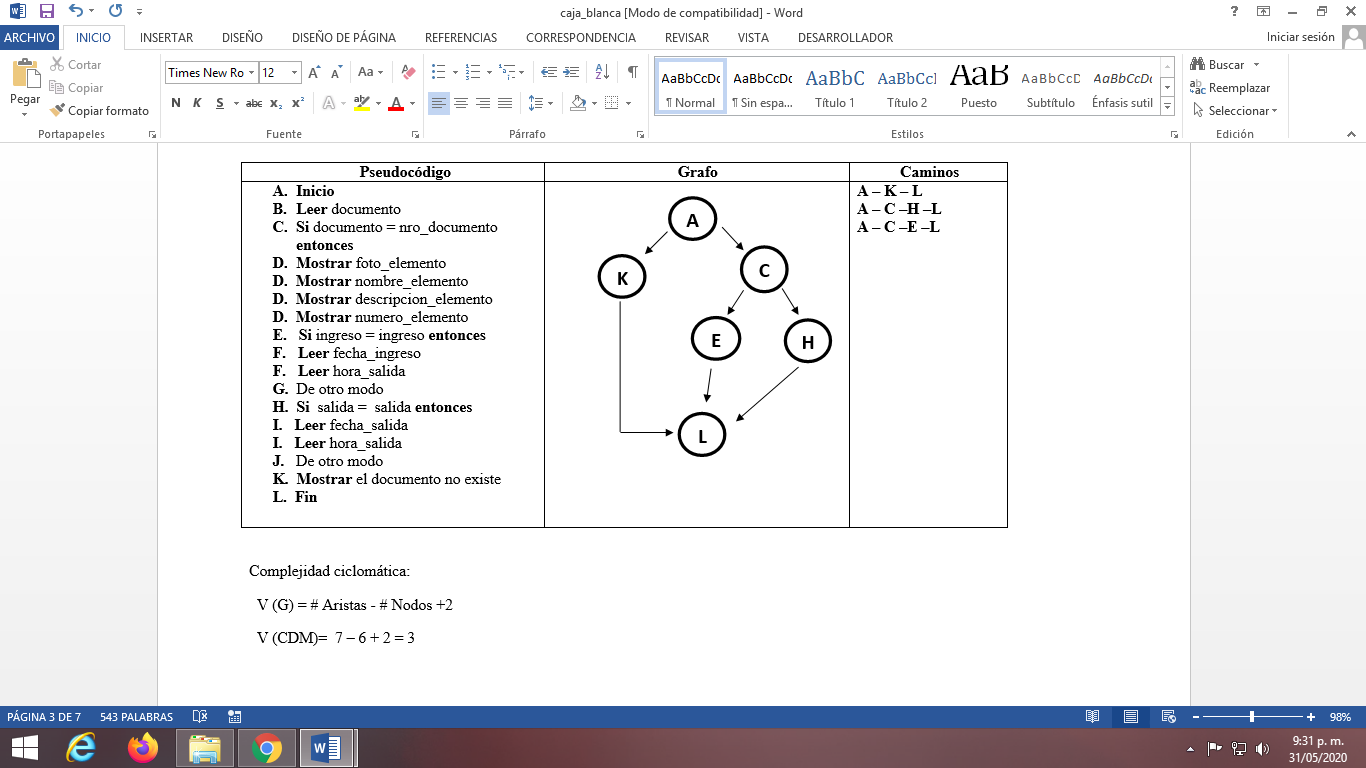
### Registrar elemento

### Editar elemento

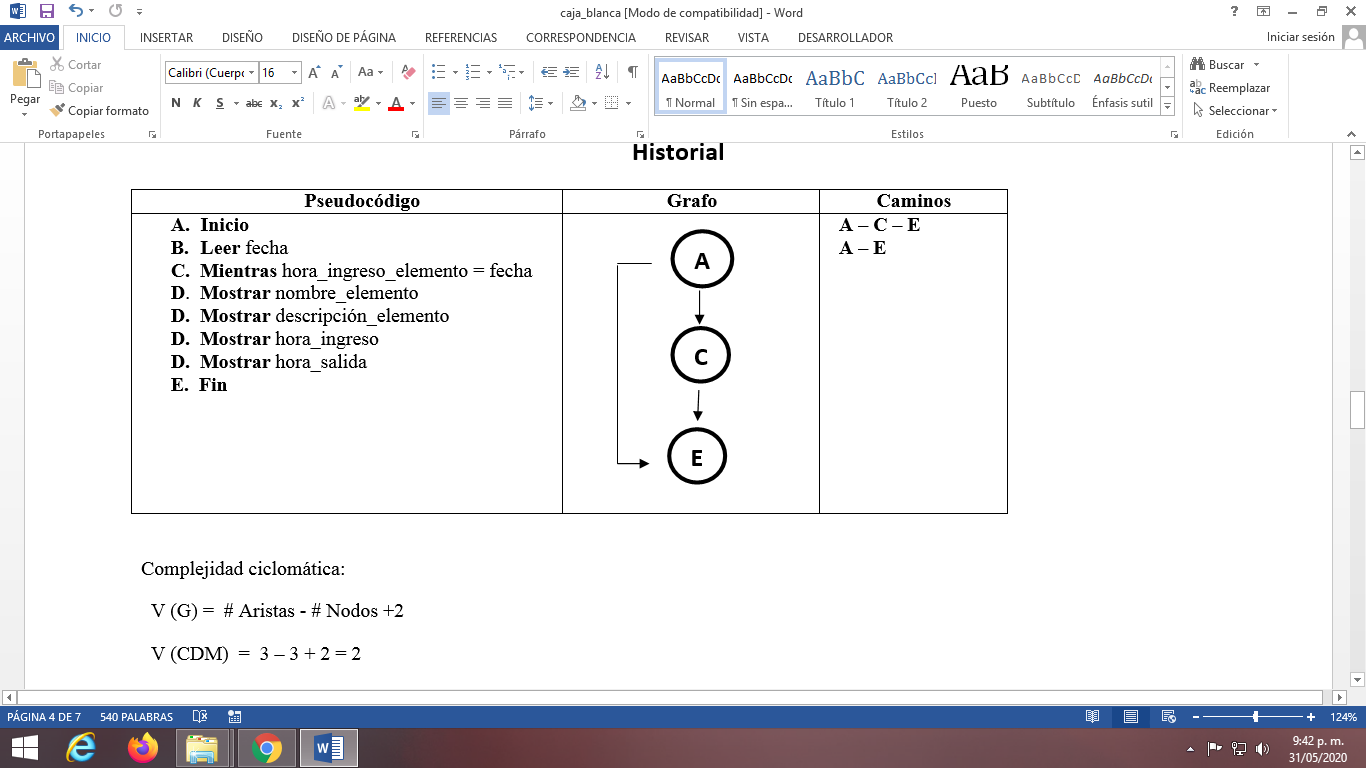


## **Modulo vigilante**

### Ingreso - salida elementos

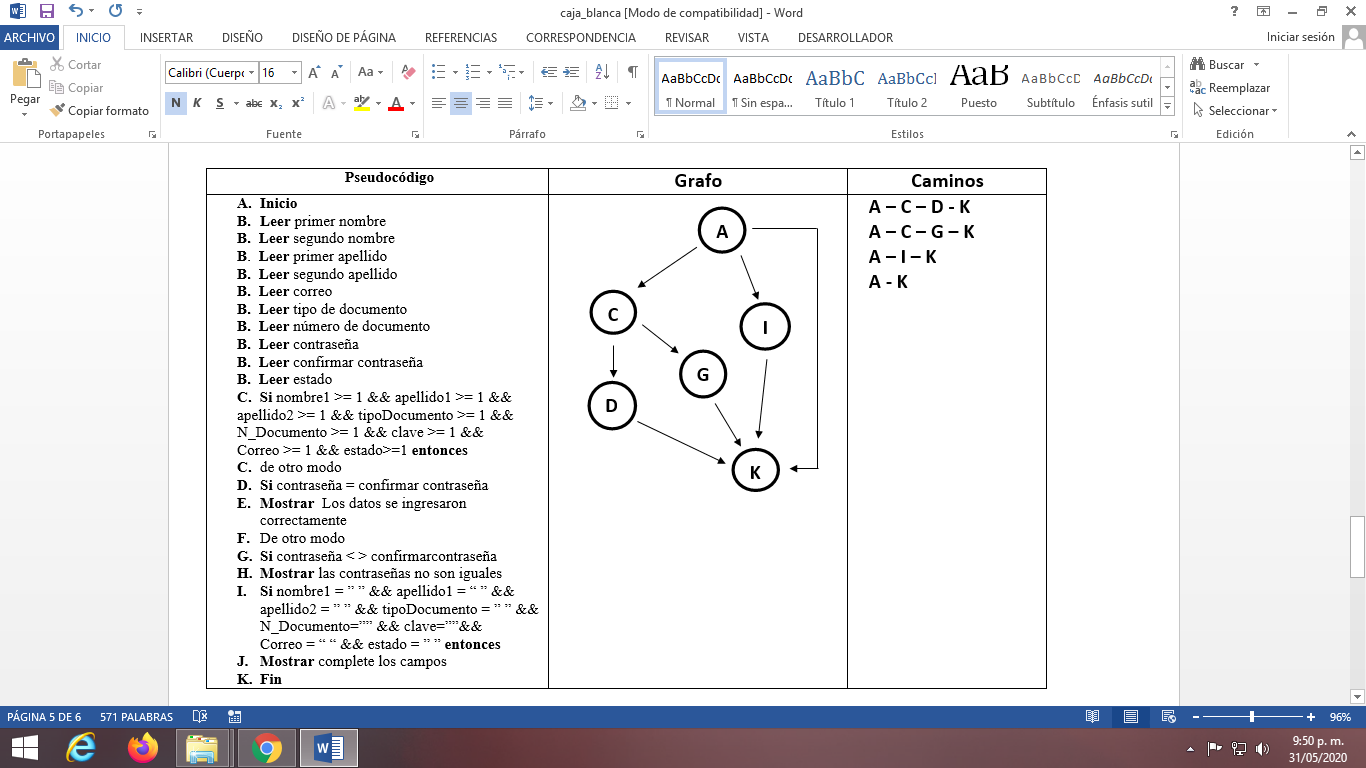


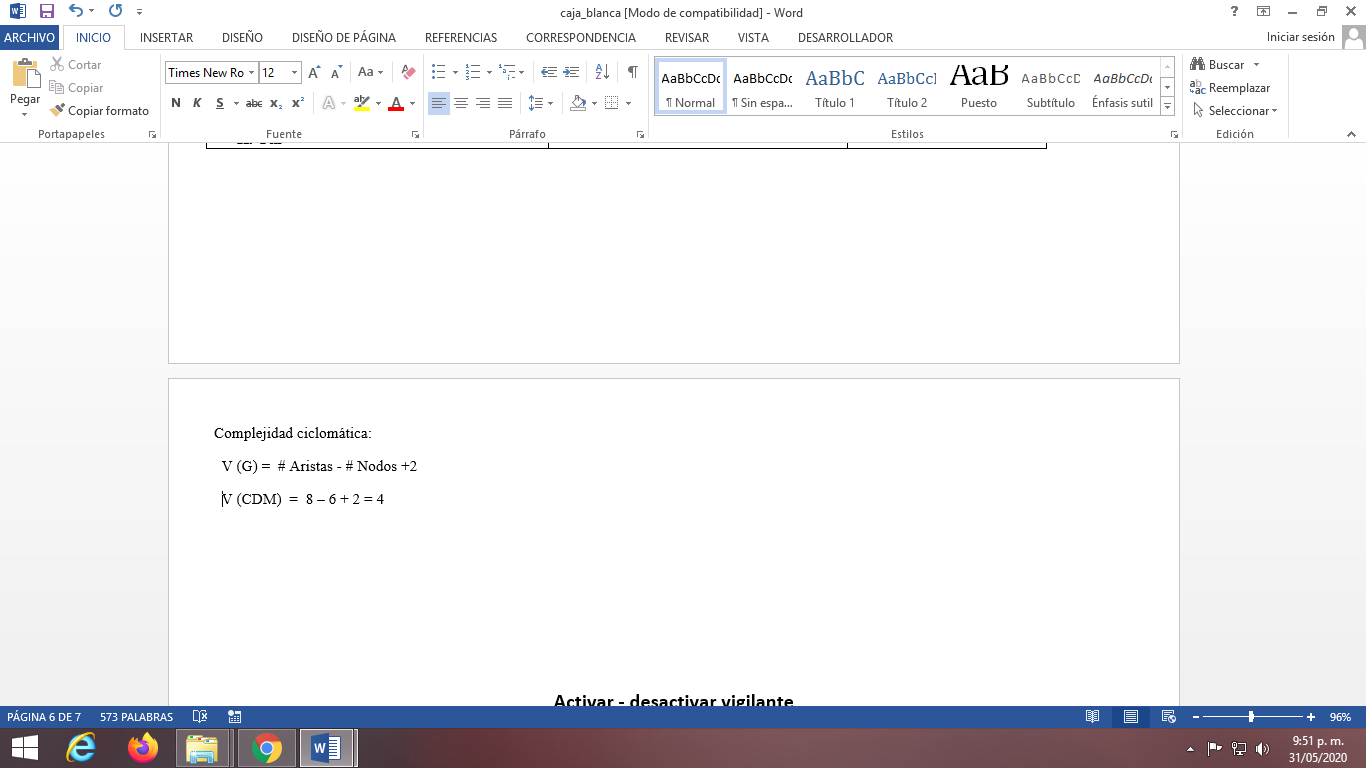
### Historial



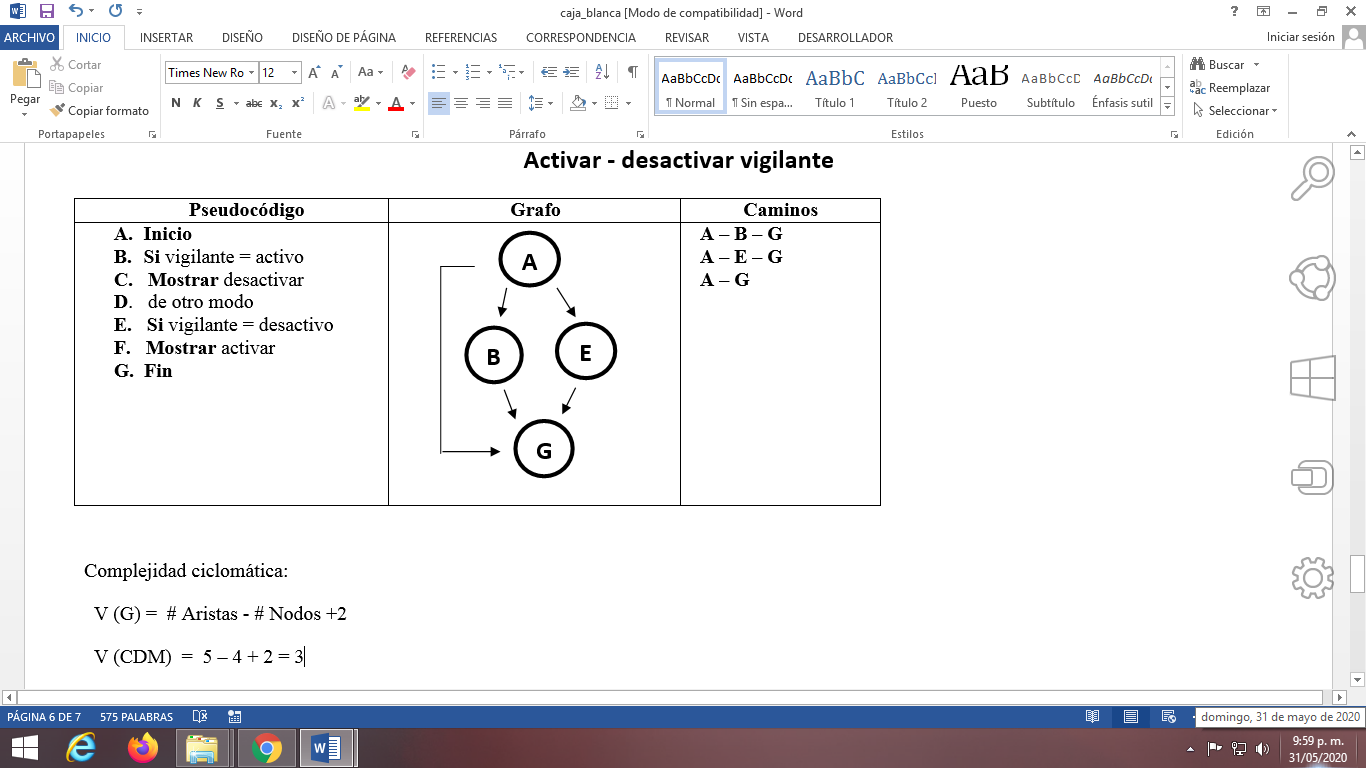
## **Modulo supervisor**

### Registrar vigilante





### Activar – desactivar vigilante



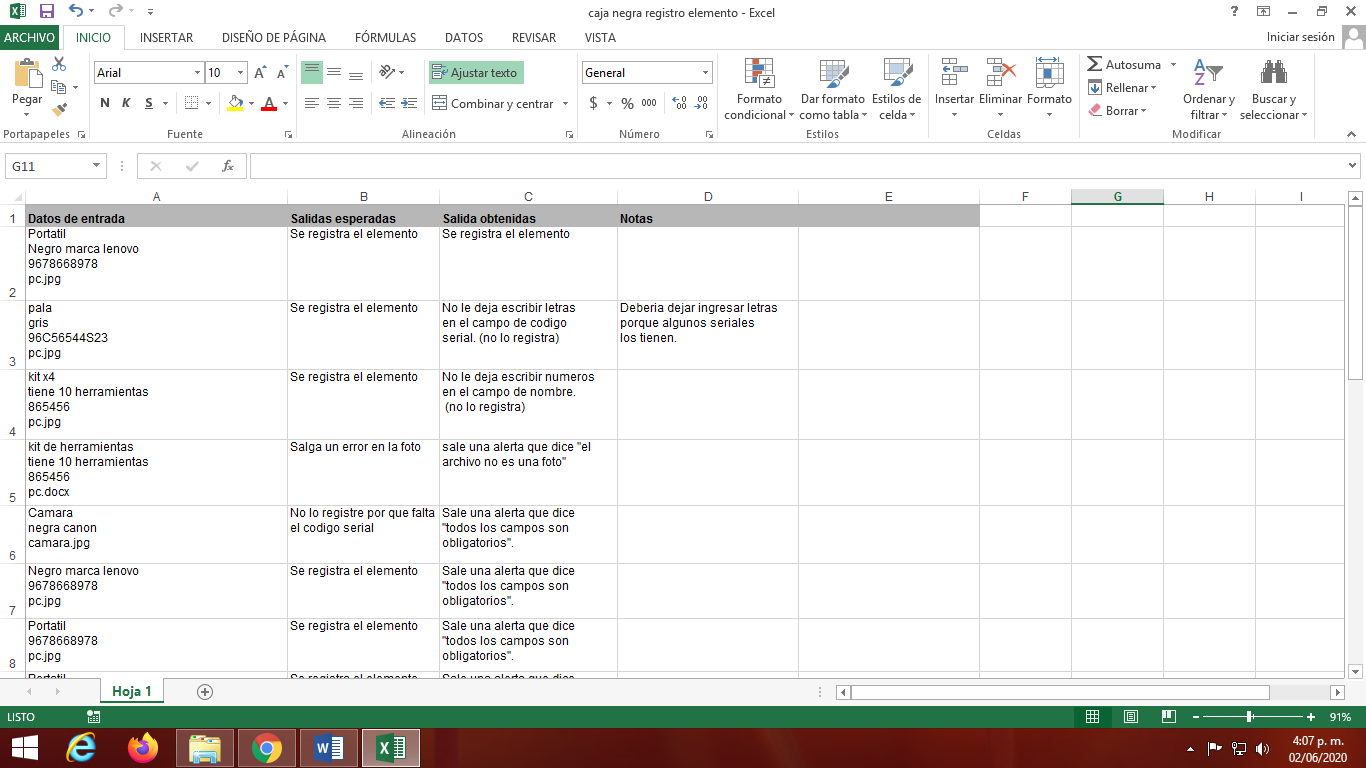
# **Pruebas de caja negra**

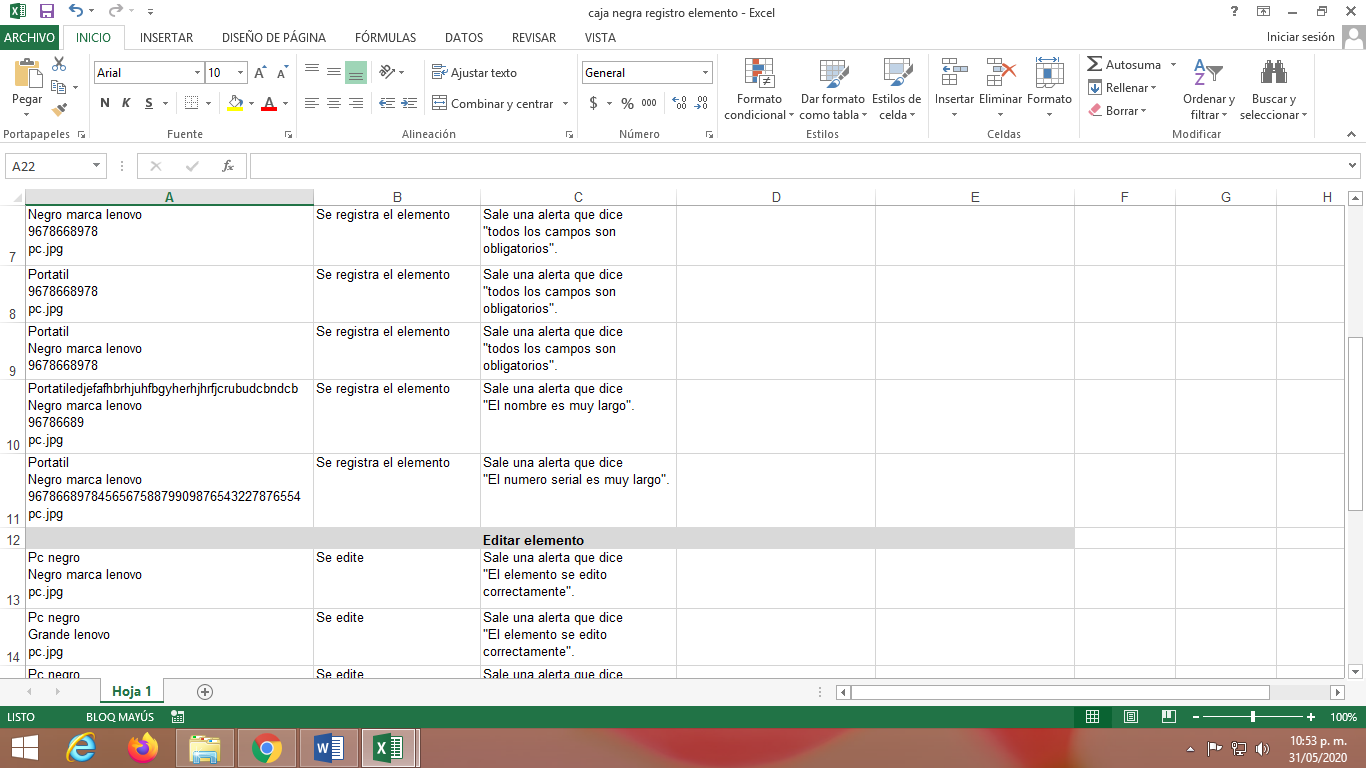
Las pruebas de caja negra se enfocan en probar la funcionalidad de un módulo específico, en base a los datos de entrada y salida, sin preocuparse en tener el conocimiento de la estructura interna del código.

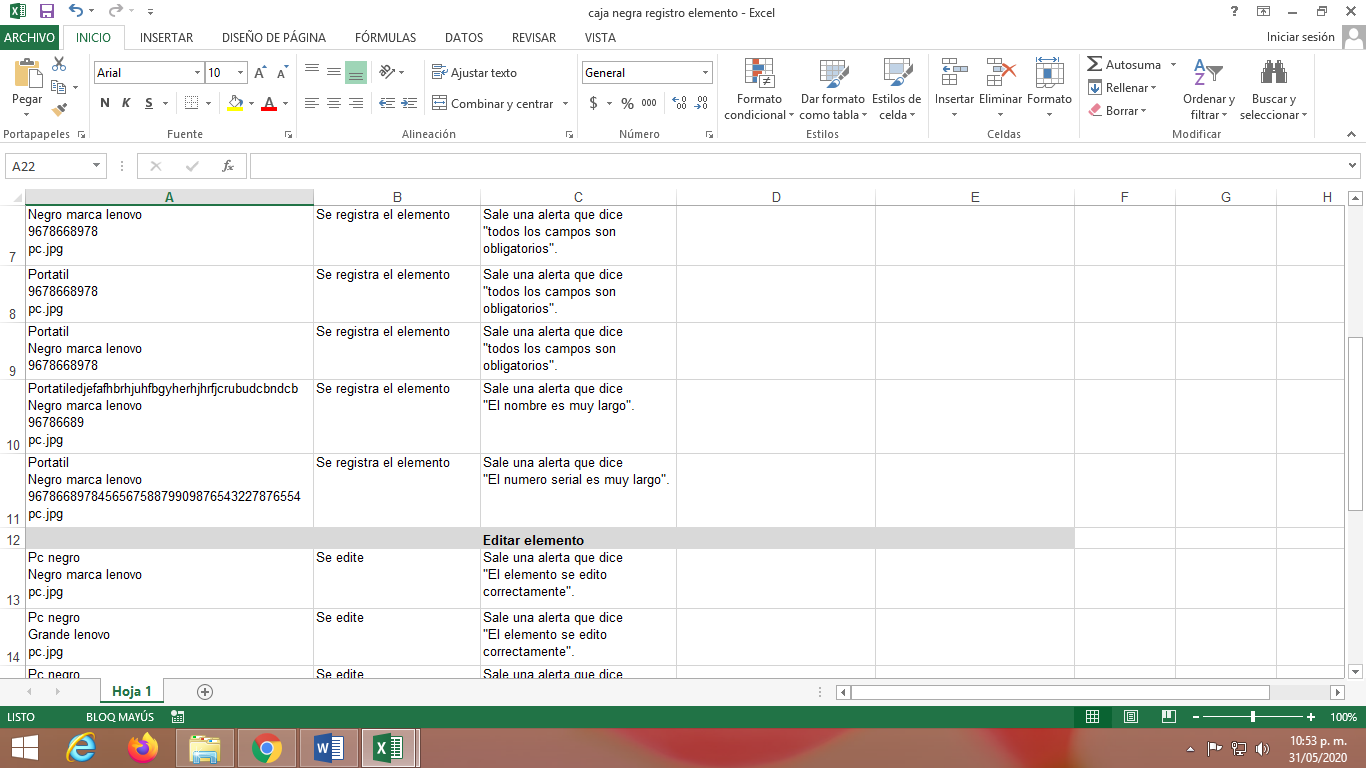
Para realizar las pruebas en el sistema ‘Registro Ágil’ se utilizó la técnica de partición equivalente, la cual consiste en encontrar un conjunto de entradas validas e invalidas, probarlo en el módulo que se va a trabajar y finalmente obtener una serie de salidas, el objetivo de esta prueba es encontrar errores en las interfaces de entrada y salida.

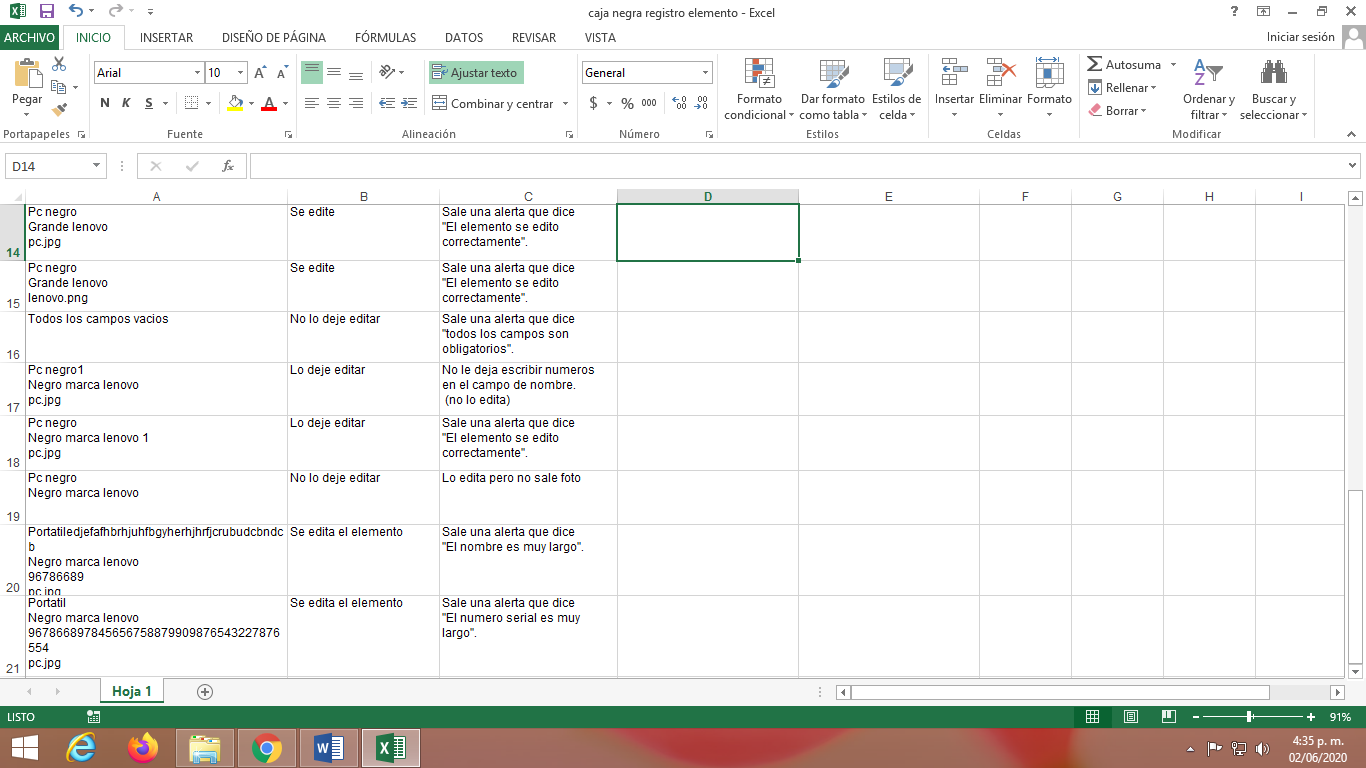
## **Modulo usurario**

### Registrar – editar elemento









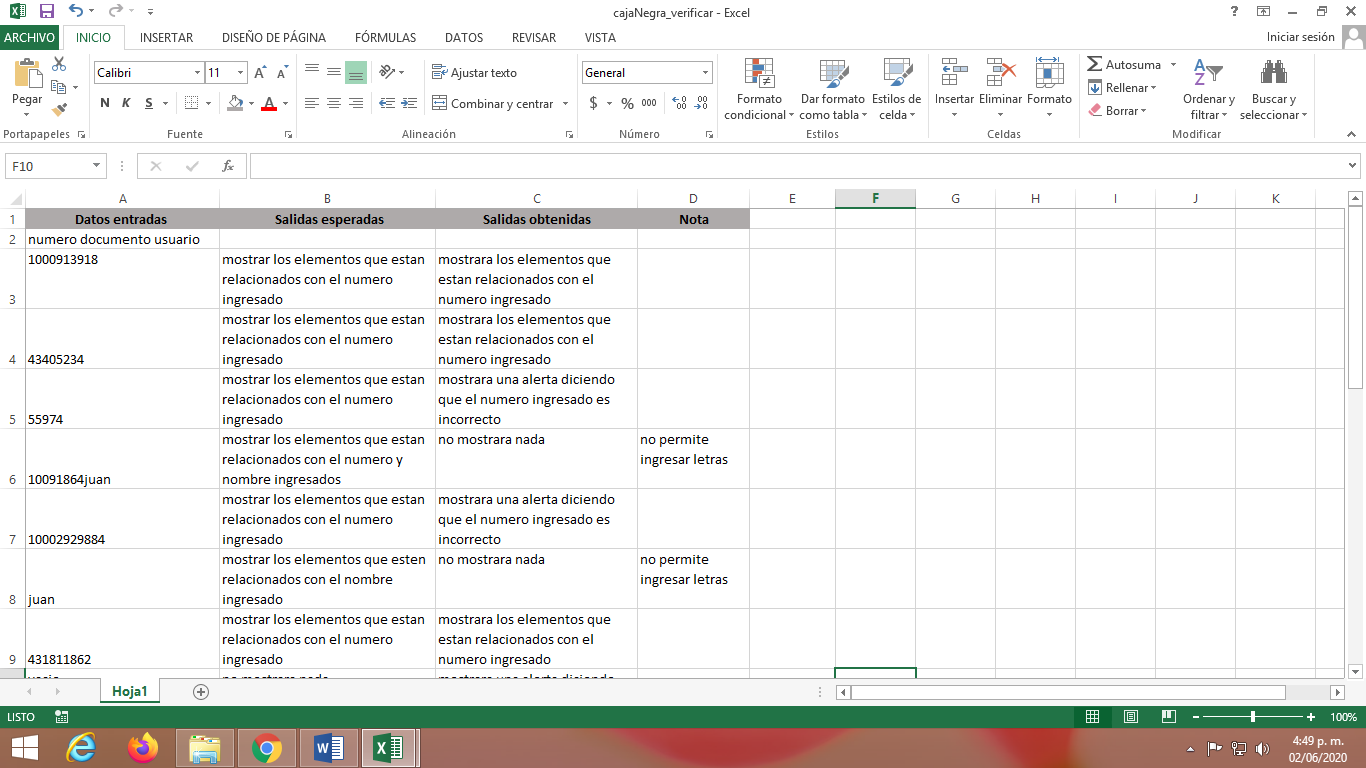
En estas pruebas se llegó a un extremo de posibles datos de entrada, dando a conocer que la mayor parte funciona como se espera y mostrando algunos errores, como lo fueron:

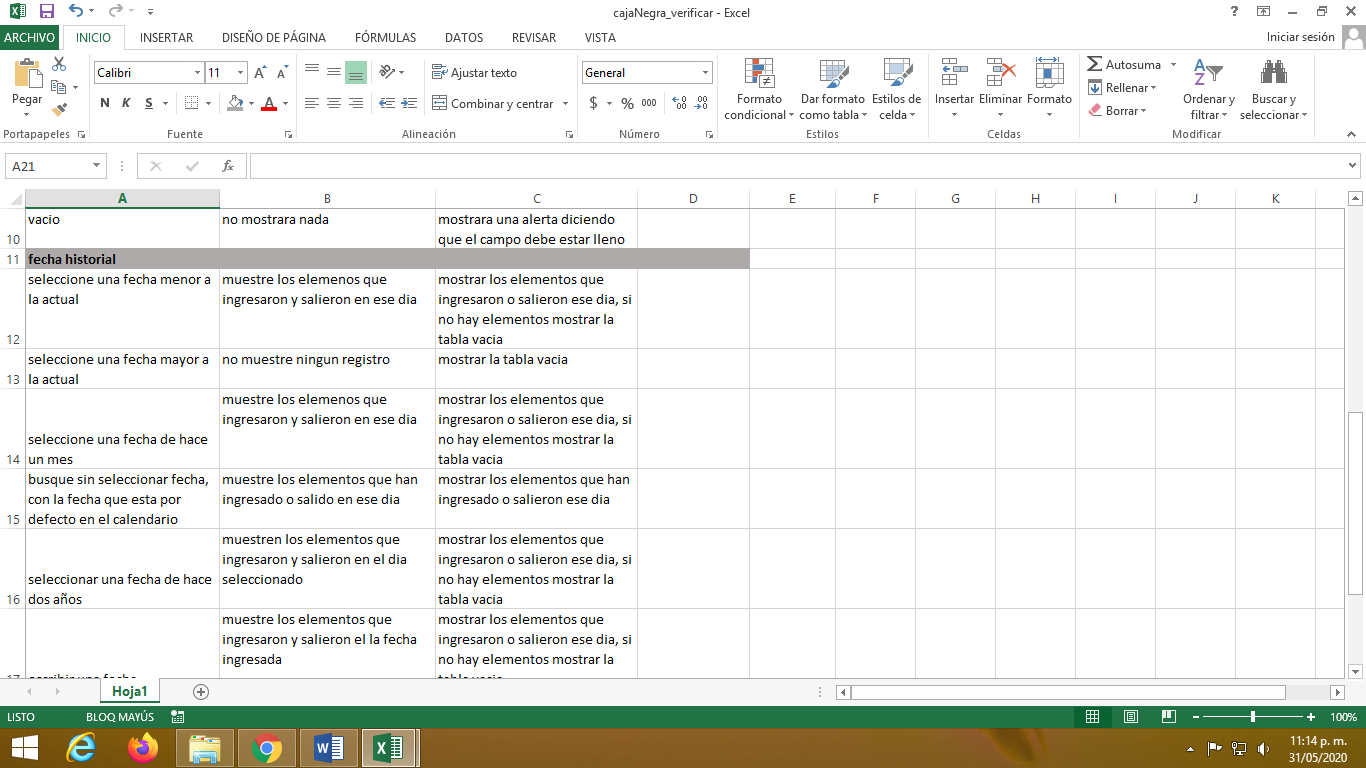
\*En el campo del número serial del elemento solo permite ingresar números, debería permitir digitar tanto número como letras ya que no sabemos que código serial tenga letras

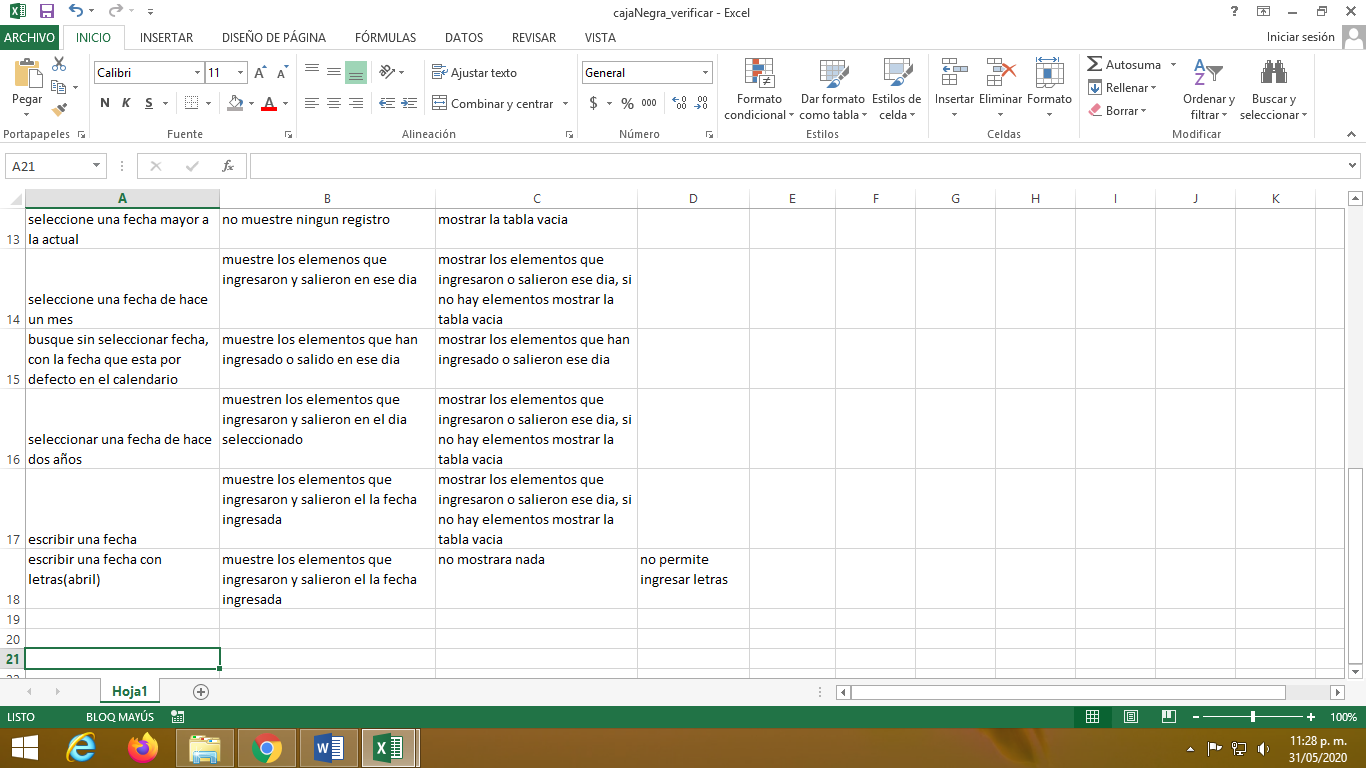
\*Al editar la información de un elemento y no sea necesario cambiar la foto, por la tanto se deja la que está, la información del elemento se edita correctamente pero no sale la foto.

## **Modulo vigilante**

### Verificación







En estas pruebas se llegó a un extremo de posibles datos de entrada, dando a conocer que la

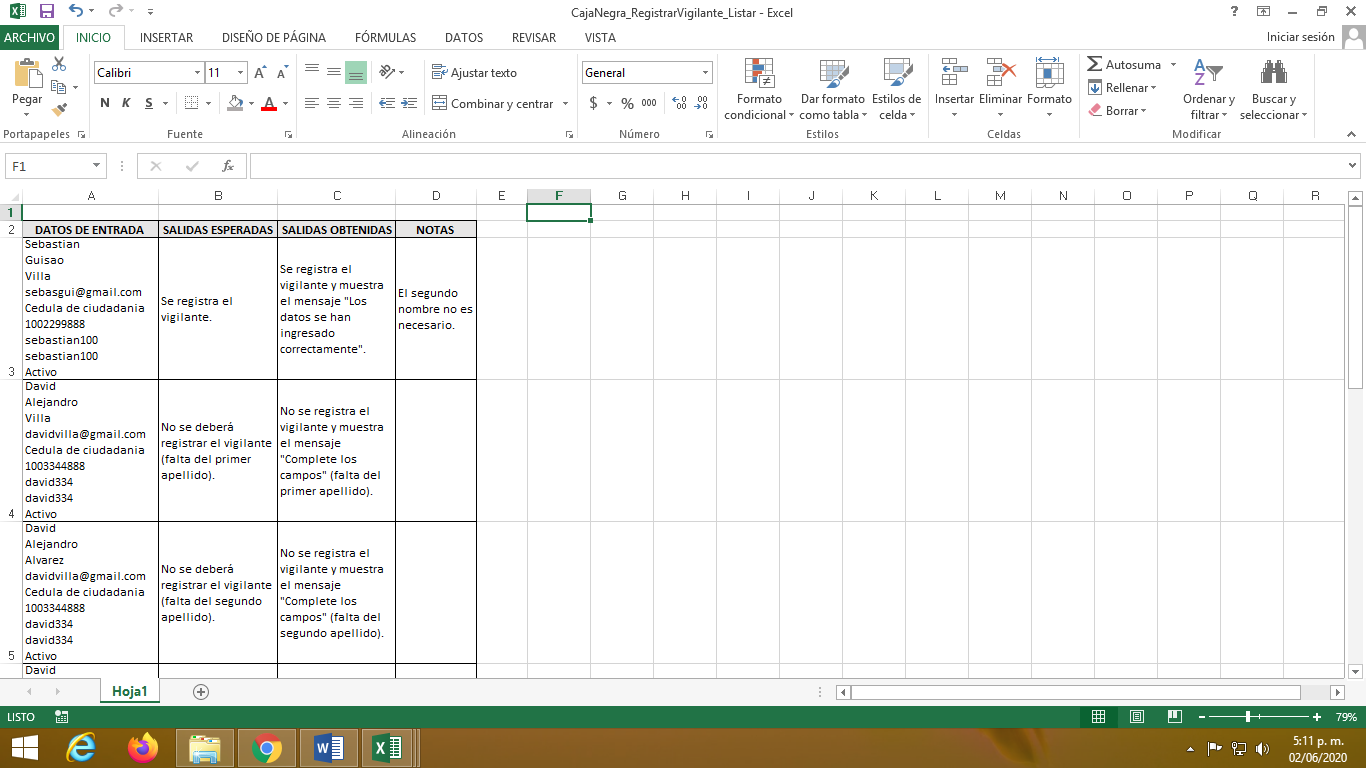
mayor parte funciona como se espera, mostrando que en el buscador del historial no permite el ingreso de letras, el número de documento tiene una cantidad de números obligatorios, en caso de no cumplir con esta cantidad saldrá una alerta informando el error, en este buscador solo se permite digitar números, si intenta ingresar alguna letra no se le permitirá, también nos da a conocer que si se ingresa un numero de documento faltándole un digito mostrara todos los elementos relacionados con dicho numero sin tomar en cuenta el digito faltante

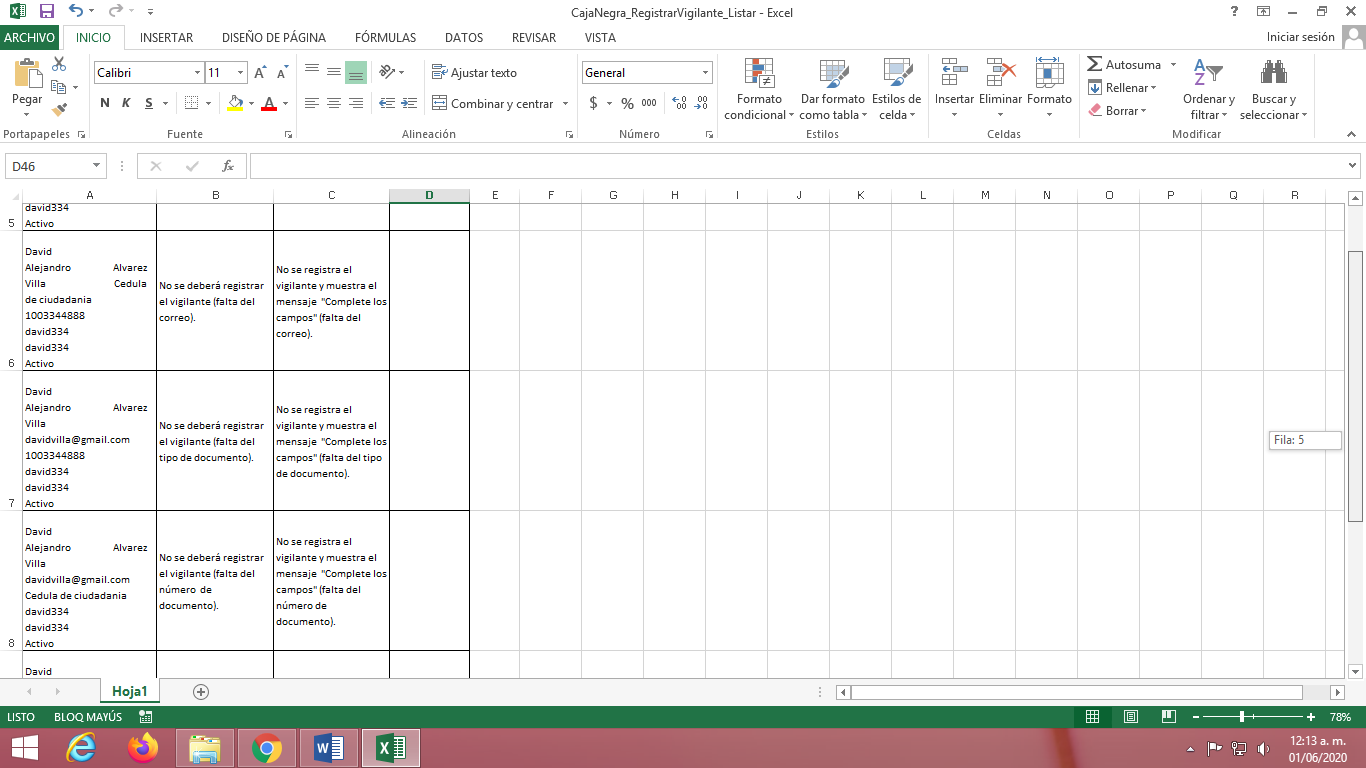
Ej. El número ingresado fue 100091391

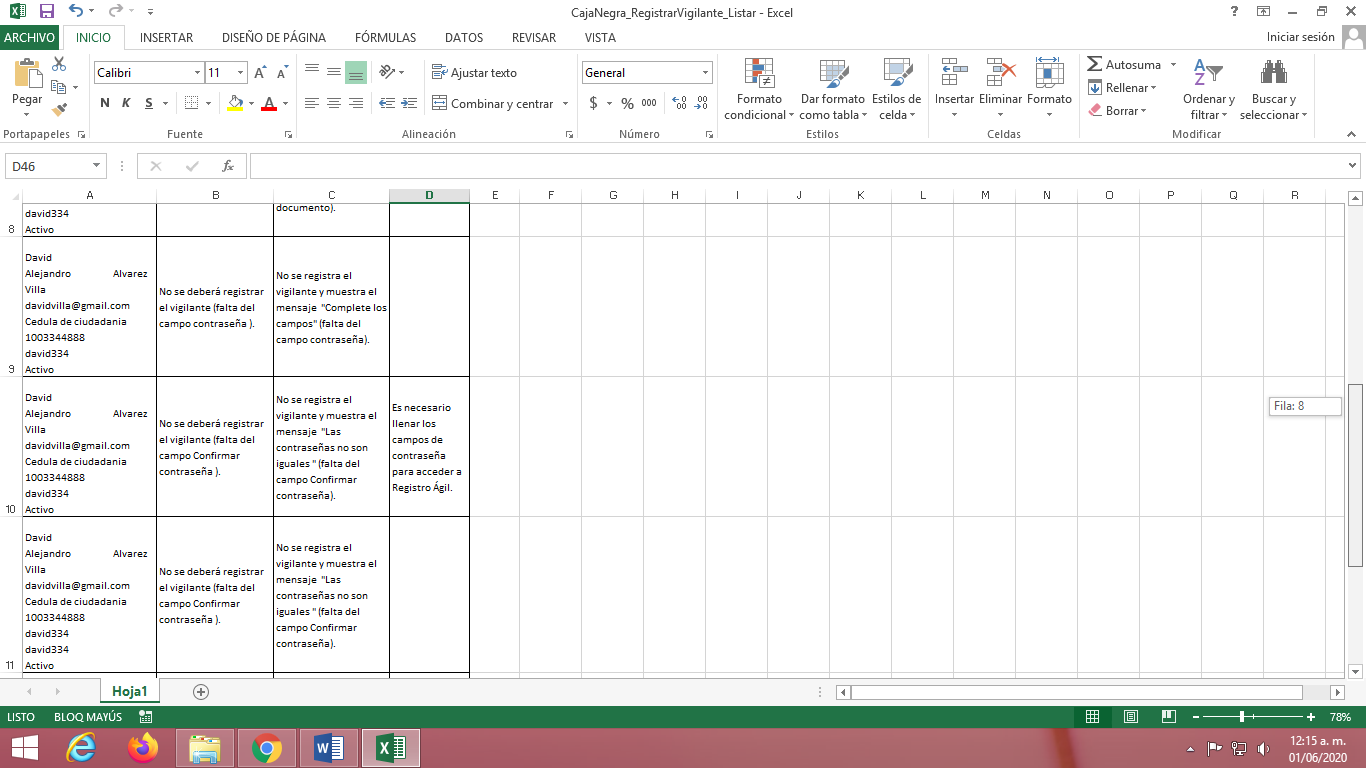
Hay elementos registrados que se relacionan con los números 1000913918, 1000913916, (traerá los elementos relacionados con los números 1000913918,1000913916)

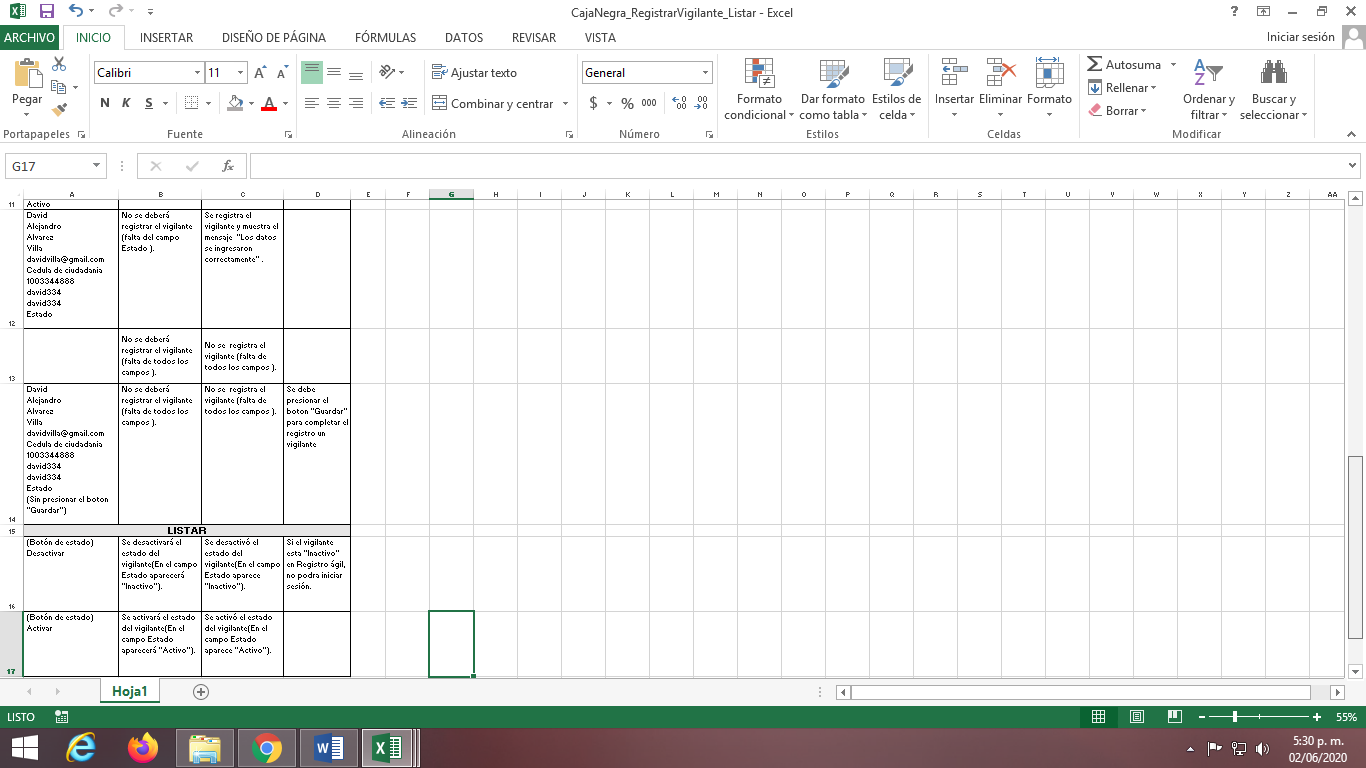
## **Modulo supervisor**

### Registrar y listar vigilante









En estas pruebas se llegó a un extremo de posibles datos de entrada, dando a conocer que para el correcto registro de un vigilante no se puede dejar ningún campo vacío, excepto el estado, si no se llena este campo se registrara correctamente el vigilante, lo cual no debería pasar ya que esta información es importante para el ingreso del vigilante a la plataforma, también se muestra que pasa cuando un vigilante está activo (puede iniciar sesión en la plataforma) o desactivo (no puede iniciar sesión en la plataforma)

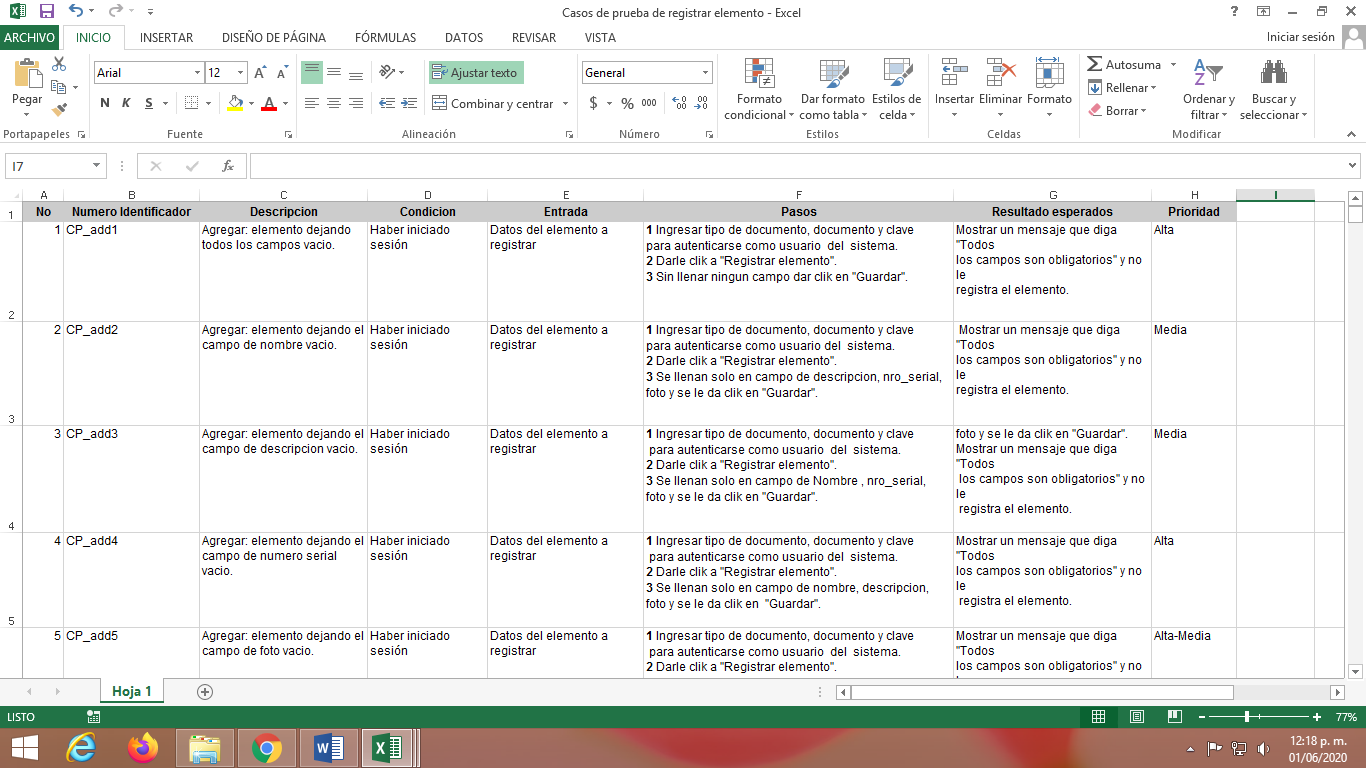
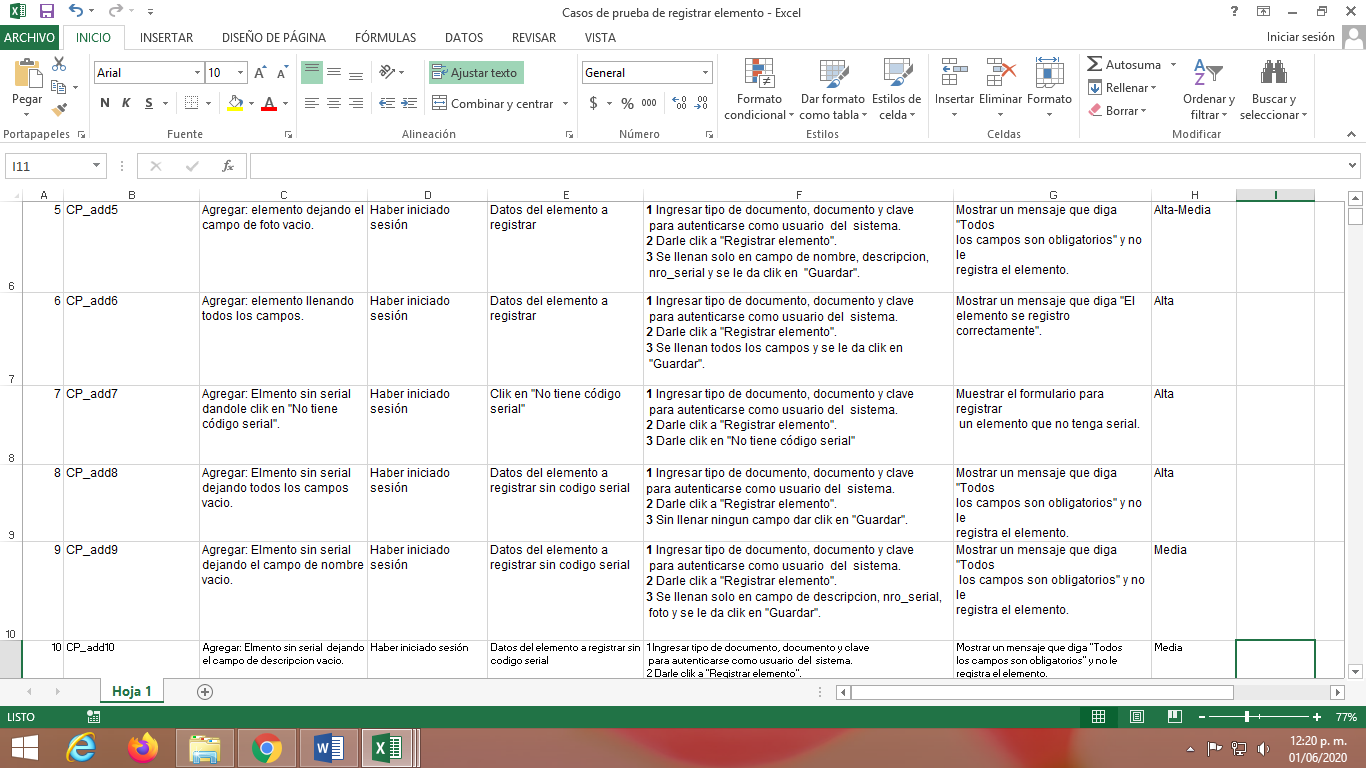
# **Casos de prueba**

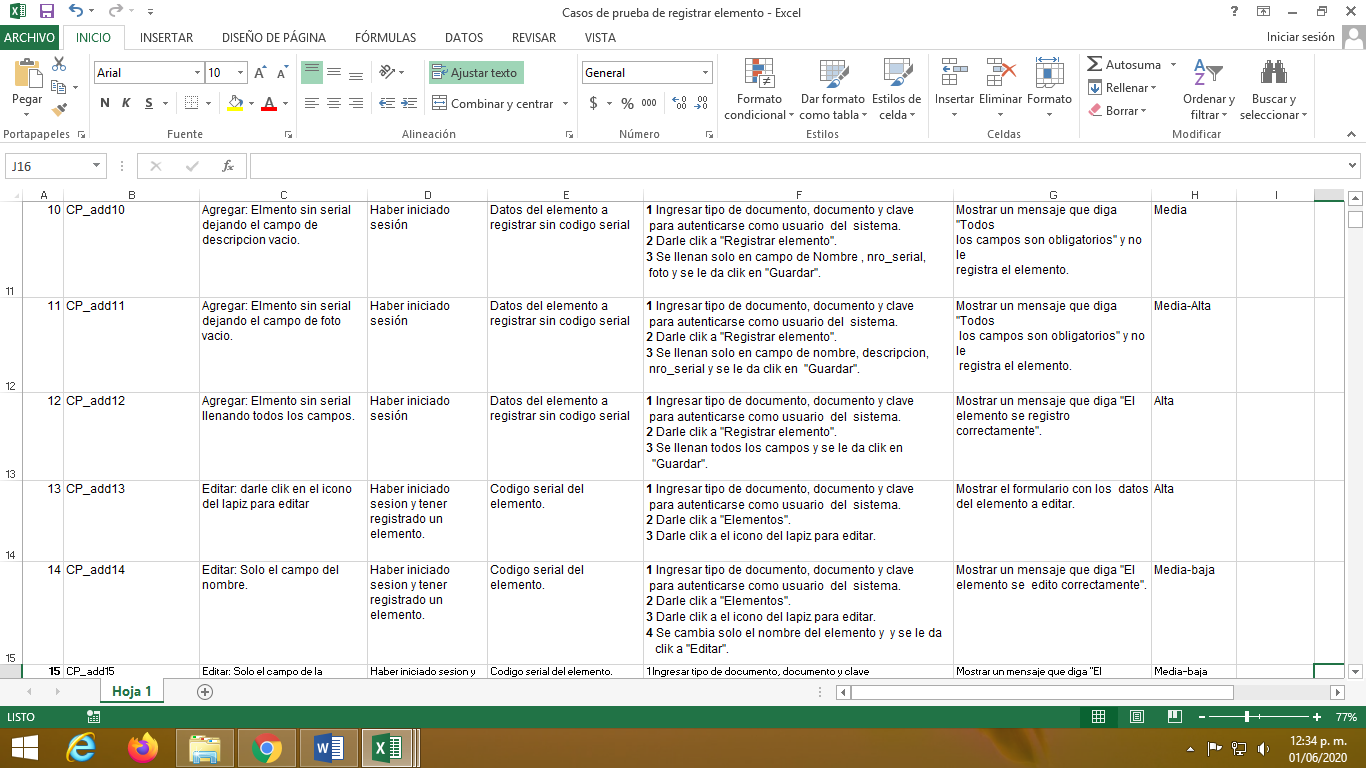
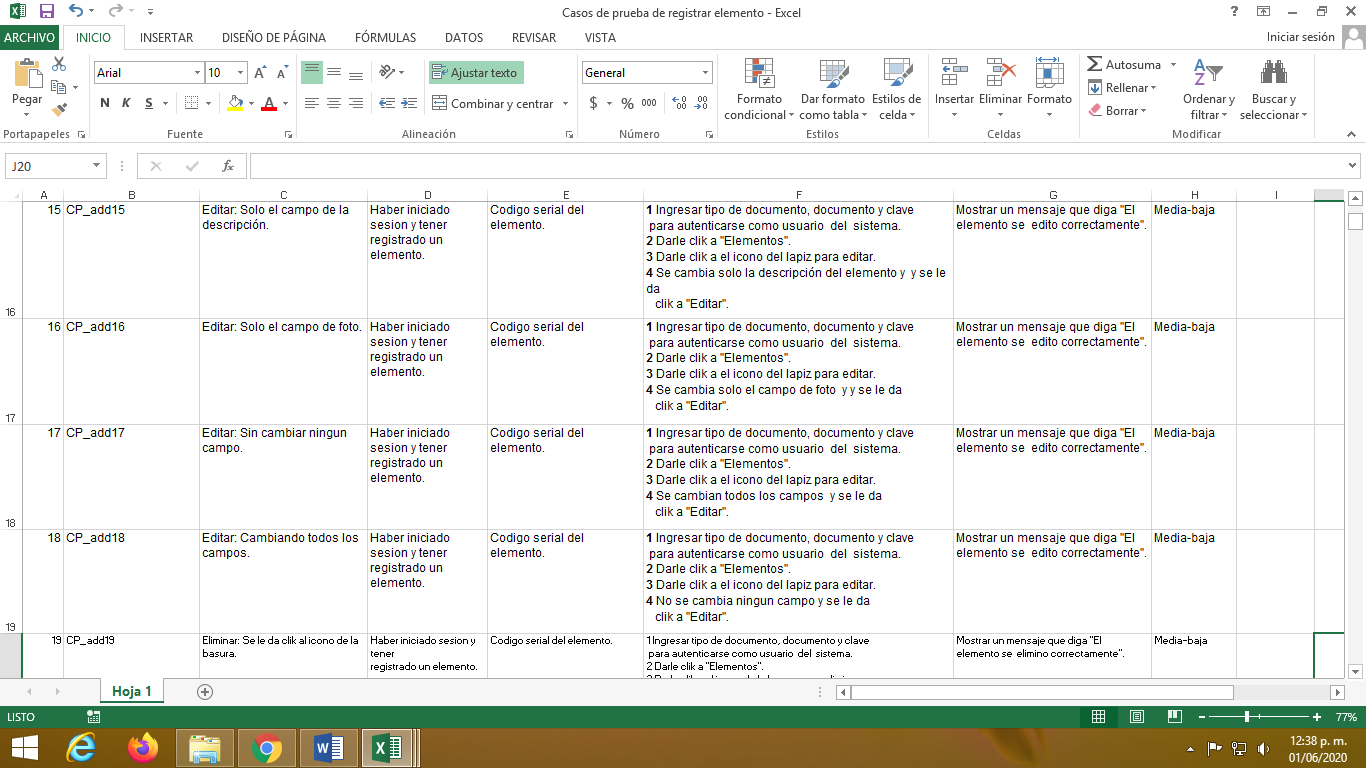
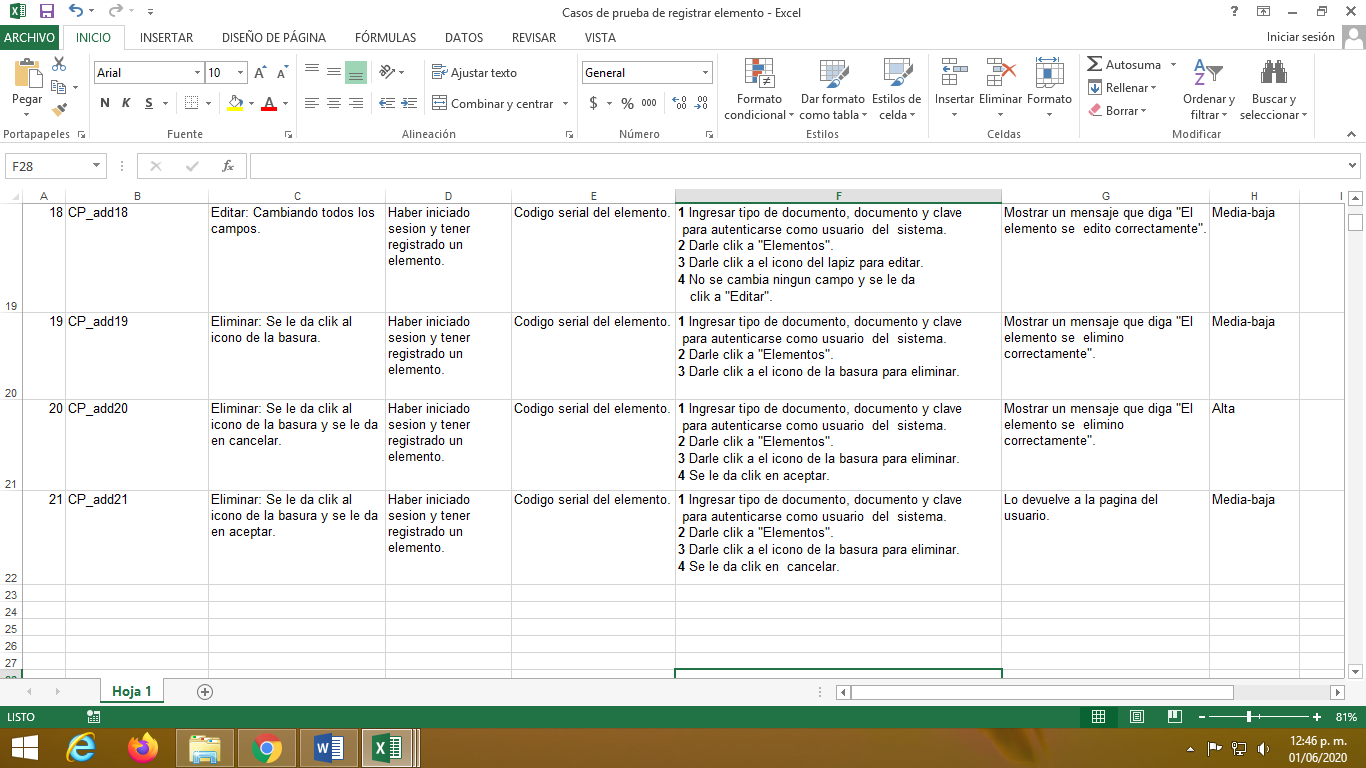
Los casos de prueba son un conjunto de valores de entrada, pre-condiciones y resultados esperados, tiene como objetivo ejercer una ruta y verificar el cumplimiento de un requisito específico en el programa.

Para desarrollar correctamente los casos de prueba en el sistema ‘Registro Ágil’ se estableció una condición sobre las funcionalidades elegidas y determinar si cumple el resultado esperado o no.

## **Modulo usuario**

### R**egistrar elemento**





En estos casos de prueba se da a conocer que para el correcto registro de los elementos no se puede dejar ningún campo vacío, ya sea que el elemento tenga número de código serial o no, muestra que se puede editar cualquier campo de un elemento, no es obligatorio editarlos todos al tiempo. Así mismo, incluye qué se debe hacer y cuáles son los pasos a seguir para realizar cada procedimiento.

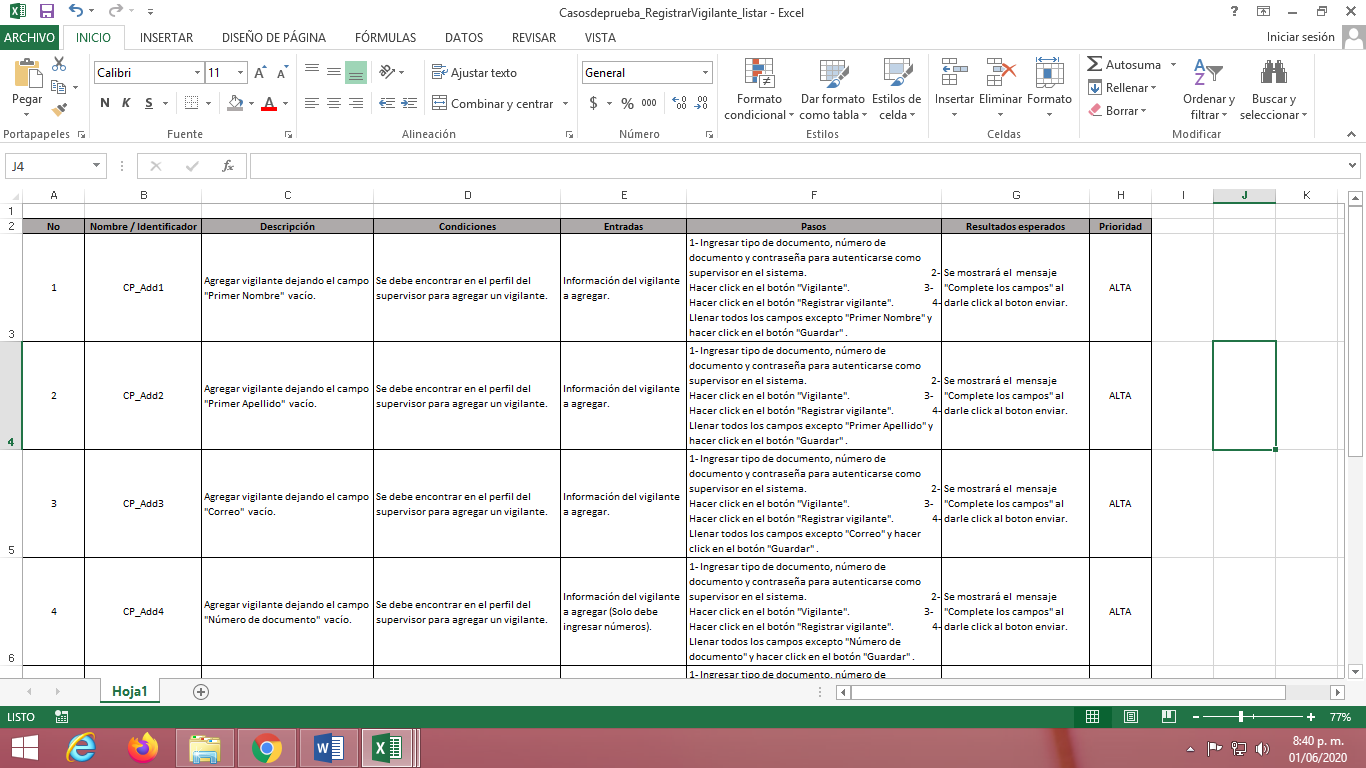
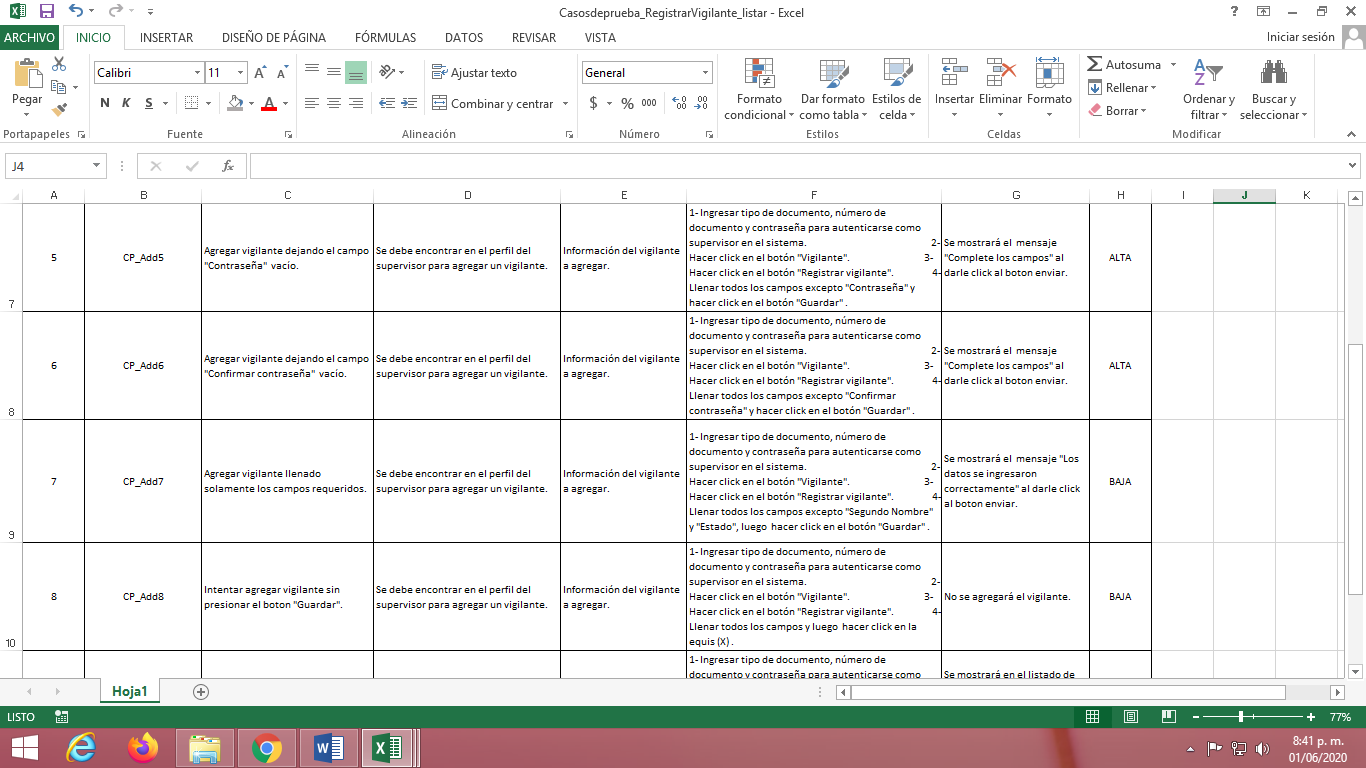
## **Modulo vigilante**

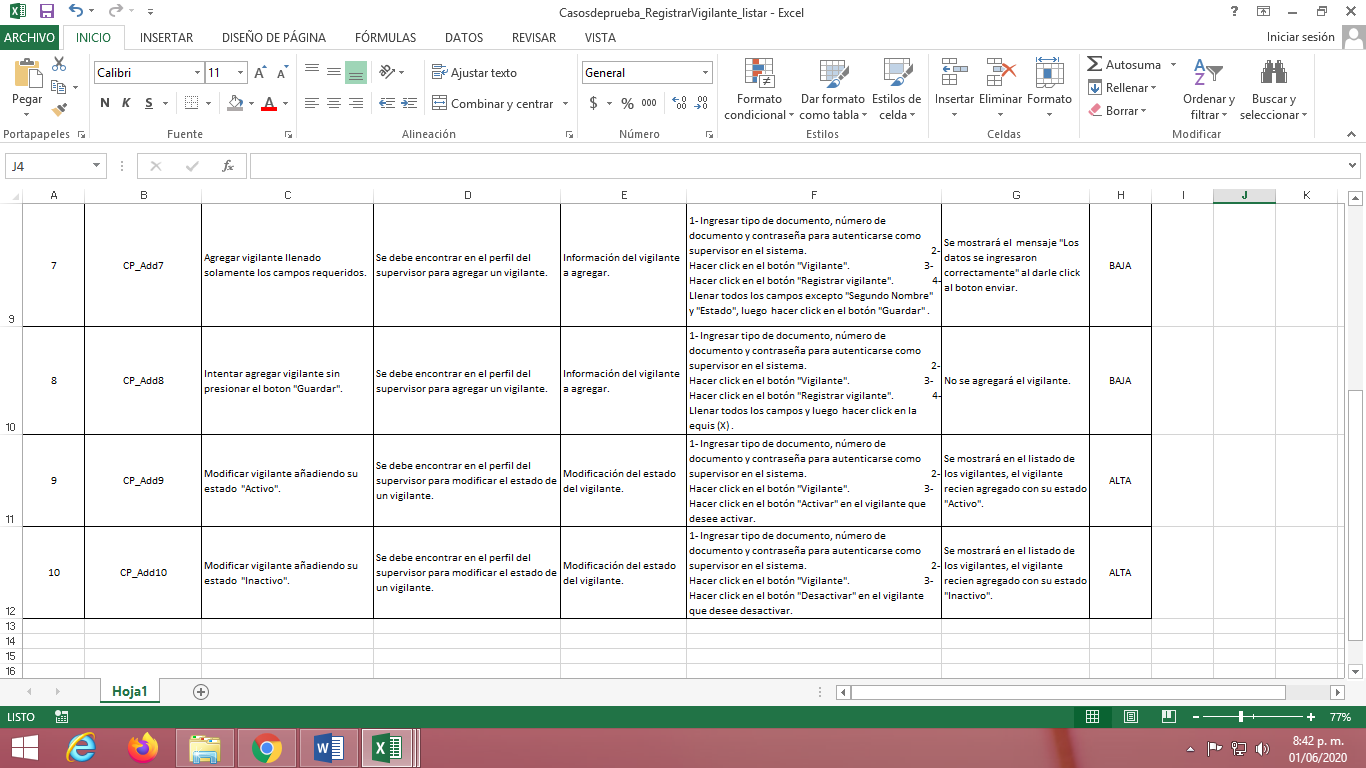
### V**erificación**

En estos casos de prueba se da a conocer cuáles son los pasos a seguir para realizar la búsqueda del historial, búsqueda de un elemento y cuáles son los posibles resultados a encontrar, muestra que pasa al darle ingreso o salida a un elemento, no permite darle ingreso a un elemento y sin darle salida volver a dar ingreso, del mismo modo no permite darle salida a un elemento y sin darle ingreso volver a dar salida.

## **Modulo supervisor**

### R**egistrar – listar vigilante**





En estos casos de prueba se da a conocer cuáles son los pasos a seguir para el registro de un vigilante, para el correcto registro se deben llenar todos los campos obligatorios, el registro no se realizara si no le da clic en guardar y se podrá cambiar el estado del vigilante a activar o desactivar (esto permitirá o no que el vigilante inicie sesión en la plataforma).

# **Conclusión**

Es muy importante realizar a nuestros proyectos las pruebas de software ya que con ellas nos damos cuenta que aunque creamos haber culminado correctamente nuestro sistema, aun nos podemos encontrar con errores que no habíamos descubierto antes, errores que pueden venir de los requerimientos, el diseño y/o la funcionalidad, de este modo, darles solución, asegurándonos que todos los módulos de nuestro proyecto van a funcionar correctamente, generando una mejor calidad y fiabilidad.