Universidad Tecnológica del Suroeste de Guanajuato

Jorge Valdez Teniente

Francisco Ismael Peña Ojeda

Jorge Eduardo Cruz Méndez

Programación de Aplicaciones

Ensaño de trabajo final

Ariana Gómez Contreras

Ingeniería en Tecnologías de la Información y la Comunicación

9 “C”

Índice

[Proyecto 1](#_Toc17115405)

[Buenas prácticas 1](#_Toc17115406)

[Propiedades 1](#_Toc17115407)

[Código 3](#_Toc17115408)

[Funcionamiento 5](#_Toc17115409)

[Figura 1 Interfaz de propiedades que muestra la distribución de elementos 1](#_Toc17115419)

[Figura 2 Interfaz principal a la que solo se puede acceder si se accede mediante login 2](#_Toc17115420)

[Figura 3 Interfaz de formulario 1 2](#_Toc17115421)

[Figura 4Interfaz de formulario 2 3](#_Toc17115422)

[Figura 5 Código de la interfaz de login 3](#_Toc17115423)

[Figura 6 Código de la interfaz de login 4](#_Toc17115424)

[Figura 7 Código para que primero se abra la ventana de login 4](#_Toc17115425)

[Figura 8 Aplicación del patrón Singleton 4](#_Toc17115426)

[Figura 9 Interfaz de login en ejecución 5](#_Toc17115427)

[Figura 10 Interfaz de login en ejecución con contraseña errónea 5](#_Toc17115428)

[Figura 11 Interfaz principal en ejecución para acceso a los formularios 6](#_Toc17115429)

[Figura 12 Interfaz de formulario 1 en ejecución 6](#_Toc17115430)

[Figura 13 Interfaz formulario 2 en ejecución 7](#_Toc17115431)

[Figura 14 Se vuelven a crear formularios 1 7](#_Toc17115432)

[Figura 15 Ejecución de patrón Singleton 7](#_Toc17115433)

# Proyecto

Es un sistema con un inicio de sesión, el cual no permitirá hacer nada en el menú principal si no se ha iniciado sesión previamente.

Donde se implementan los patrones de diseño Singleton y MVC, los cuales nos ayudaron al momento de la organización y funcionamiento de los diferentes archivos del sistema permitiéndonos así llevar un mejor control de su funcionalidad.

# Buenas prácticas

* Aplicación de la sangría: al momento de iniciar una nueva línea de código para identificar más fácilmente donde inicia una nueva sentencia.
* Alineación de las llaves: para identificar donde se abre una llave y donde se cierra, en caso de que falte una llave, identificar cual es rápidamente.
* Nombre de los archivos: para identificar más fácilmente que hace cada archivo con respecto de los otros.
* Comentar el código: para tener una idea más clara de que hace cada fragmento de código.
* Se cuidaron aspectos del diseño: se implementaron fondos simples para una mejor vista, además de acomodarse los elementos del diseño de forma distribuida.
* Diseño intuitivo: se creó un diseño intuitivo para que cualquier persona pueda usarlo sin problema.

# Propiedades

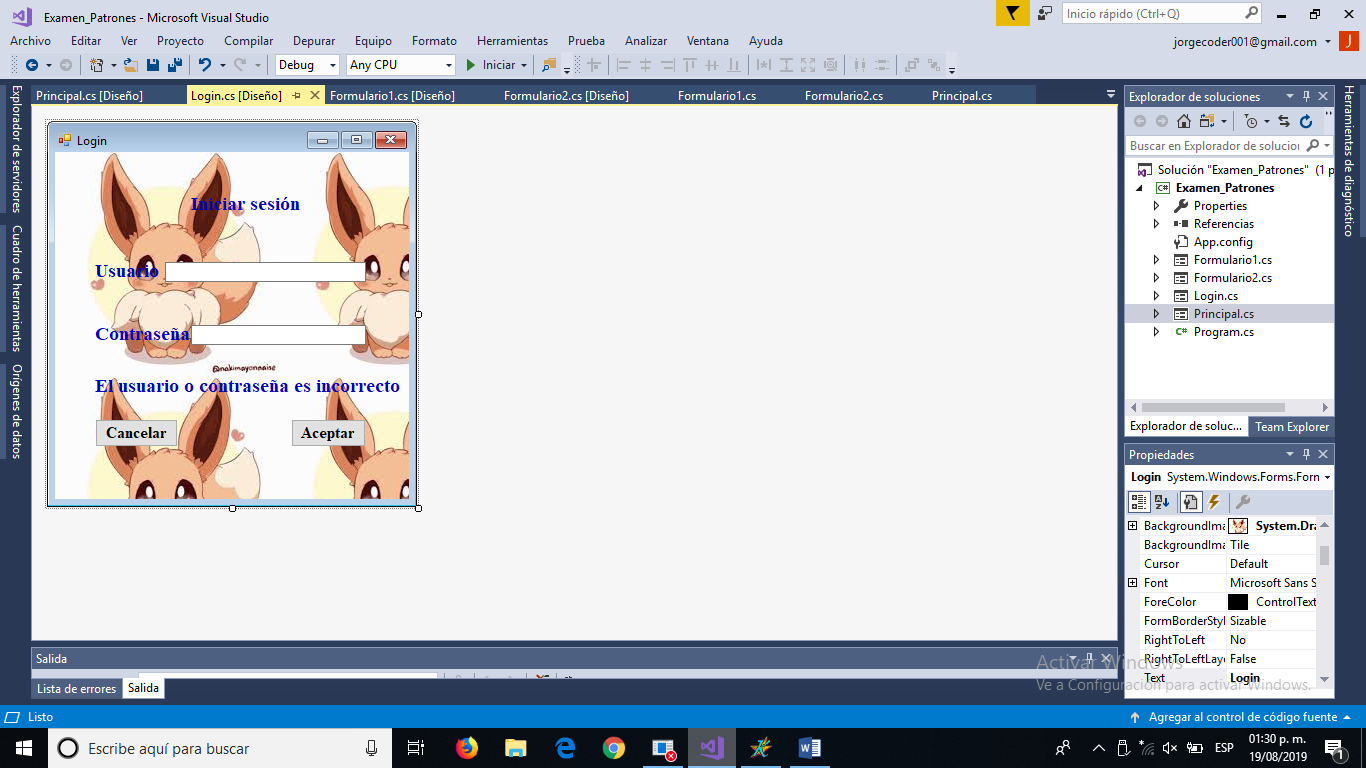


Figura Interfaz de propiedades que muestra la distribución de elementos

En la ventana de login, se cambian las siguientes propiedades:

* En la etiqueta de mensaje de error, se cambia la propiedad de Visible, para que se oculte al momento de ejecutarse.
* La propiedad de botón cancelar, se asignó el botón correspondiente a cancelar, para que termine con la ejecución del programa.

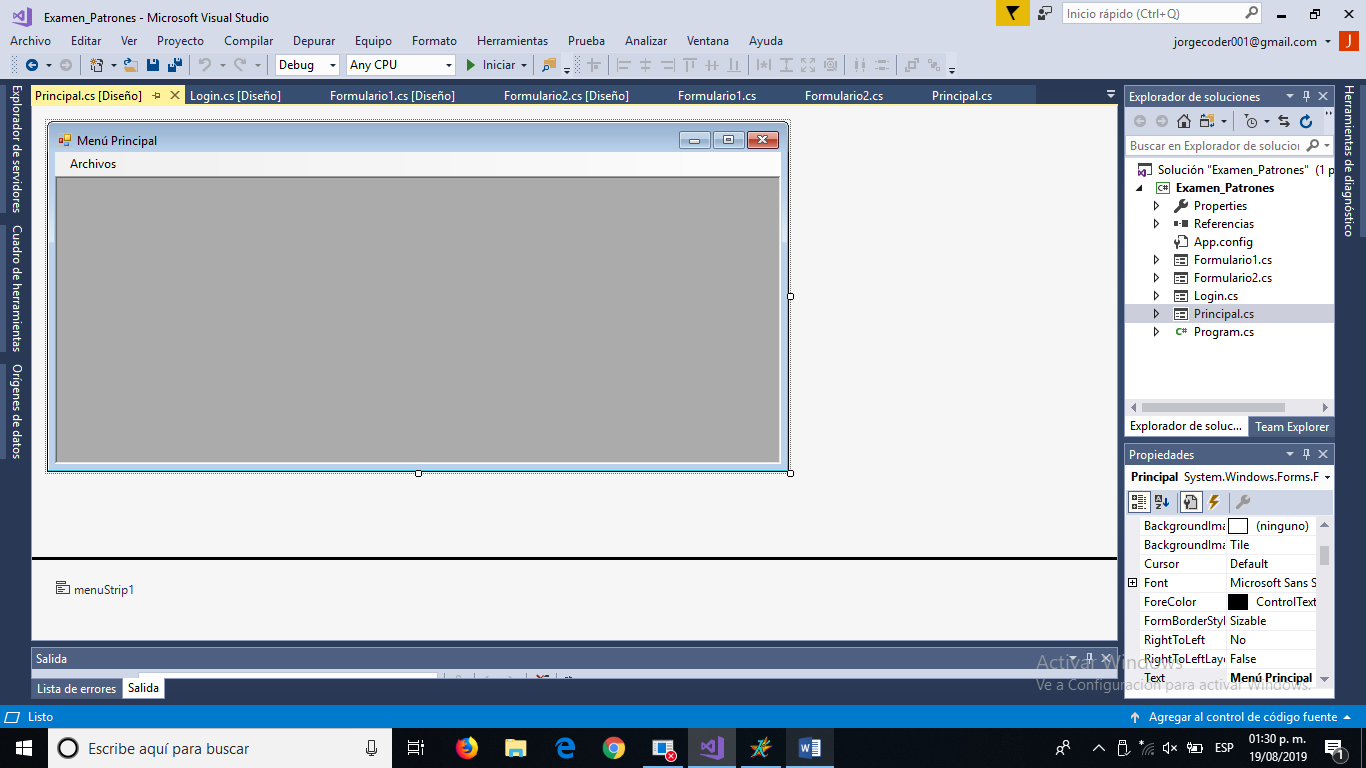


Figura Interfaz principal a la que solo se puede acceder si se accede mediante login

En la ventana principal, se cambian las siguientes propiedades:

* Windows state a maximizar, para que al ejecutarse el programa, aparezca del tamaño de la pantalla.
* ISMDIController, se cambia a verdadero, para que en este formulario, se puedan abrir nuevos formularios.

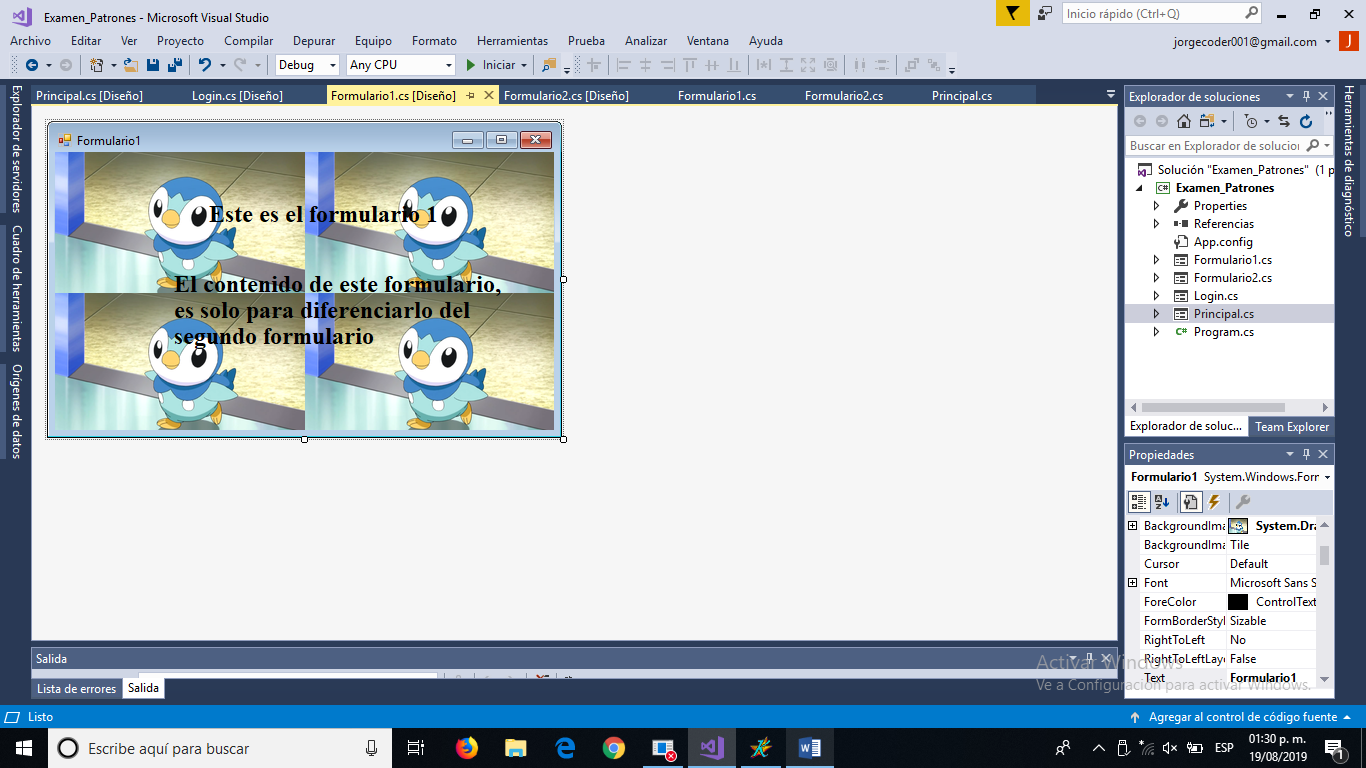


Figura Interfaz de formulario 1

En la ventana de Formulario 1, no se cambian propiedades, solo se crea un diseño diferente al de la ventana del Formulario 2.

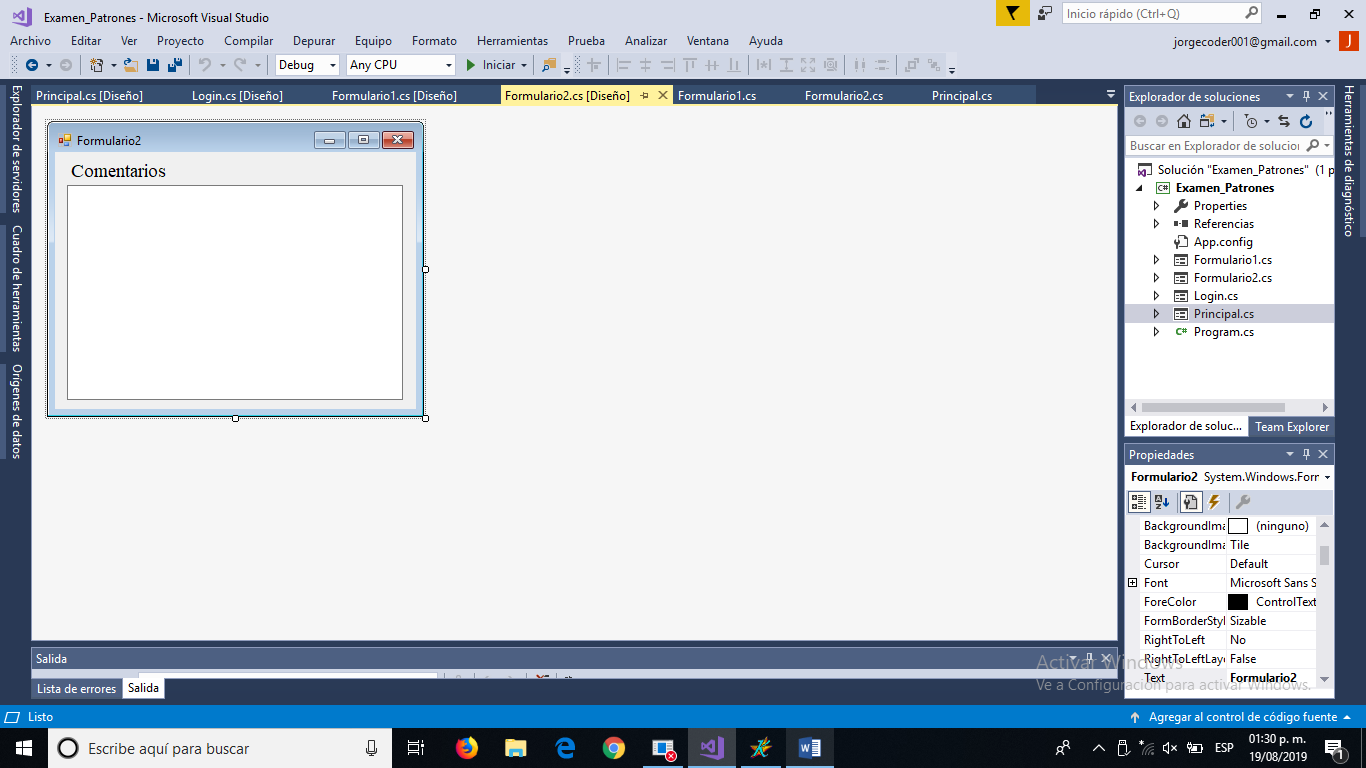


Figura Interfaz de formulario 2

En el Formulario 2, no se cambian propiedades, solo se crea un diseño diferente al del formulario 1, además de agregarle el código del patrón de diseño Singleton para su uso posterior.

# Código

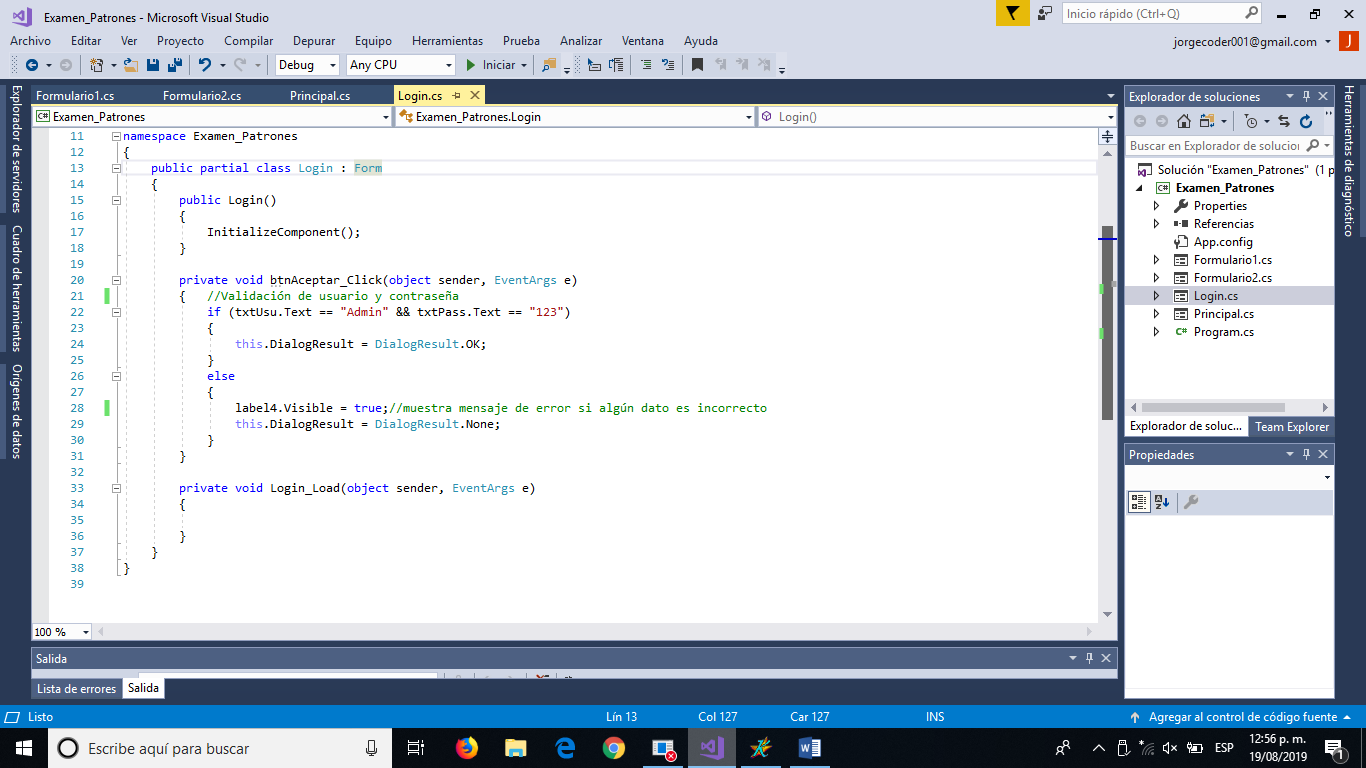


Figura Código de la interfaz de login

El código anterior, se colocó en el botón Aceptar de la interfaz de login, y lo que hace, es tomar los valores ingresados en las cajas de texto de usuario y contraseña y compararlos con los que ya están definidos, en caso de ser correctos, validará el acceso a la siguiente ventana y cerrará la de login.

En caso de ingresar algún dato erróneo, la aplicación mostrará la etiqueta con el mensaje de error y no permitirá el acceso a la aplicación.

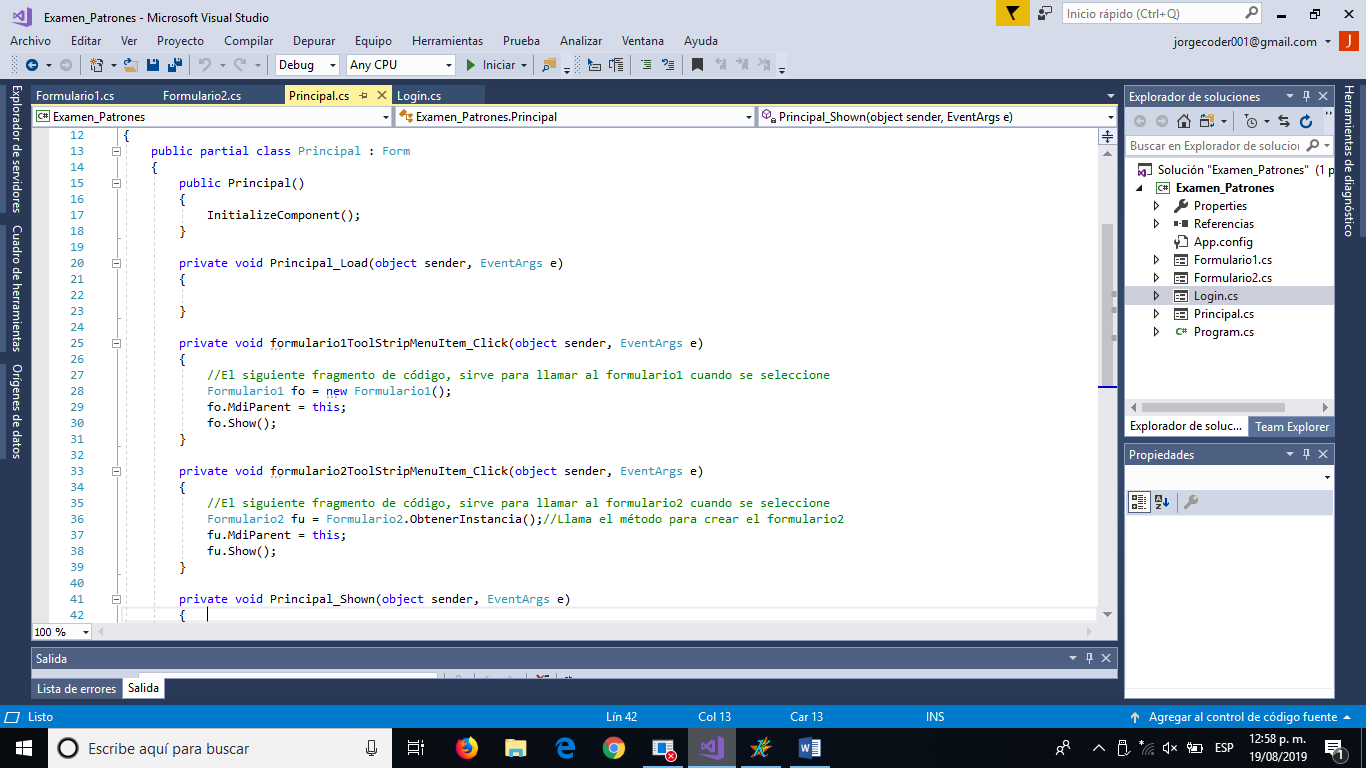


Figura Código de la interfaz de login

El fragmento de código anterior, se agregó en la interfaz principal, danto click en el menú que dice archivos y seleccionando las dos opciones de Formulario 1 y Formulario 2.

Y de lo que se encargará es de una vez seleccionada esa opción, mostrará el formulario correspondiente al seleccionado.

Sin embargo, el Formulario 2, se llama un método que se explicará más adelante.

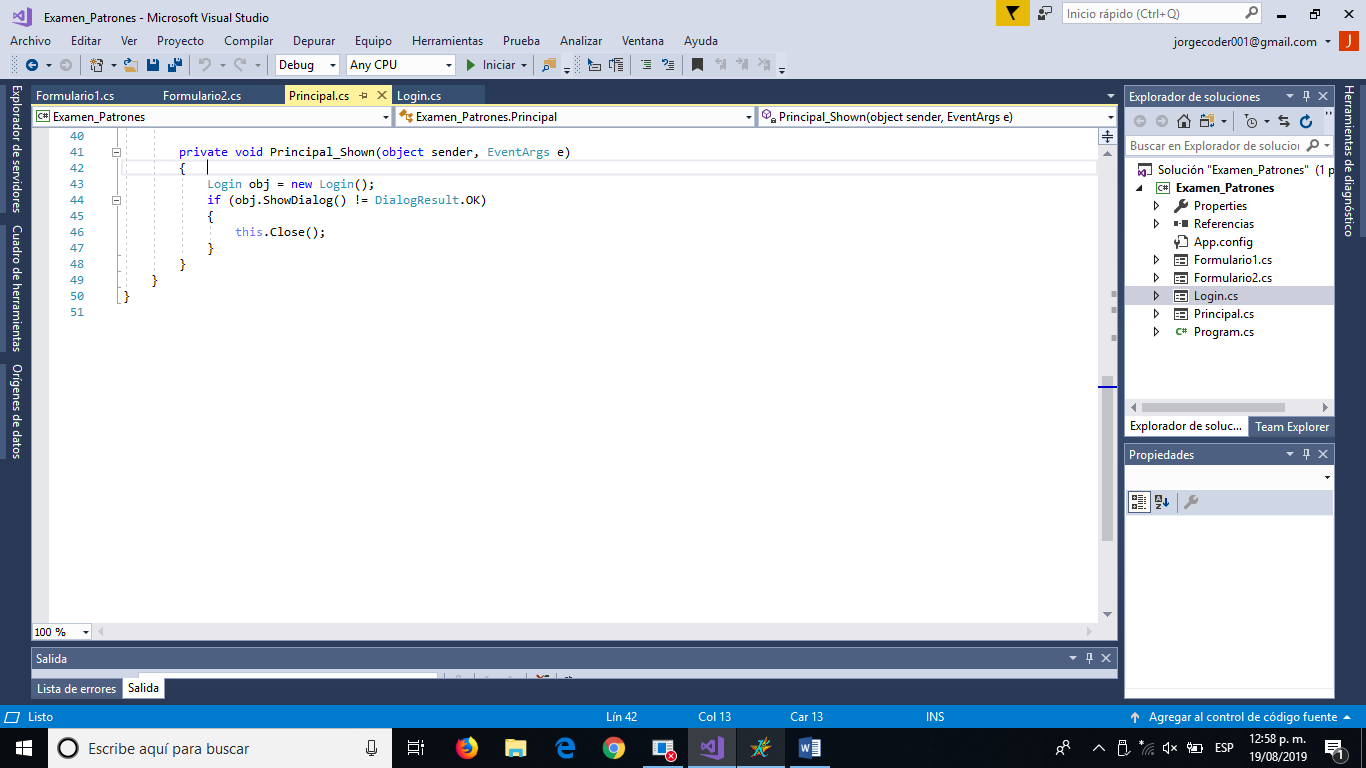


Figura Código para que primero se abra la ventana de login

También en la interfaz principal, se agregó un evento de tipo “Shown”, donde se instanció la ventana de login, para que una vez se ejecute el programa, se muestre la ventana principal y delante se muestre el login, el cual impedirá el uso de la ventana principal hasta que no se haya iniciado sesión.

Y si se cierra la ventana de login sin iniciar sesión, detendrá la ejecución del programa.

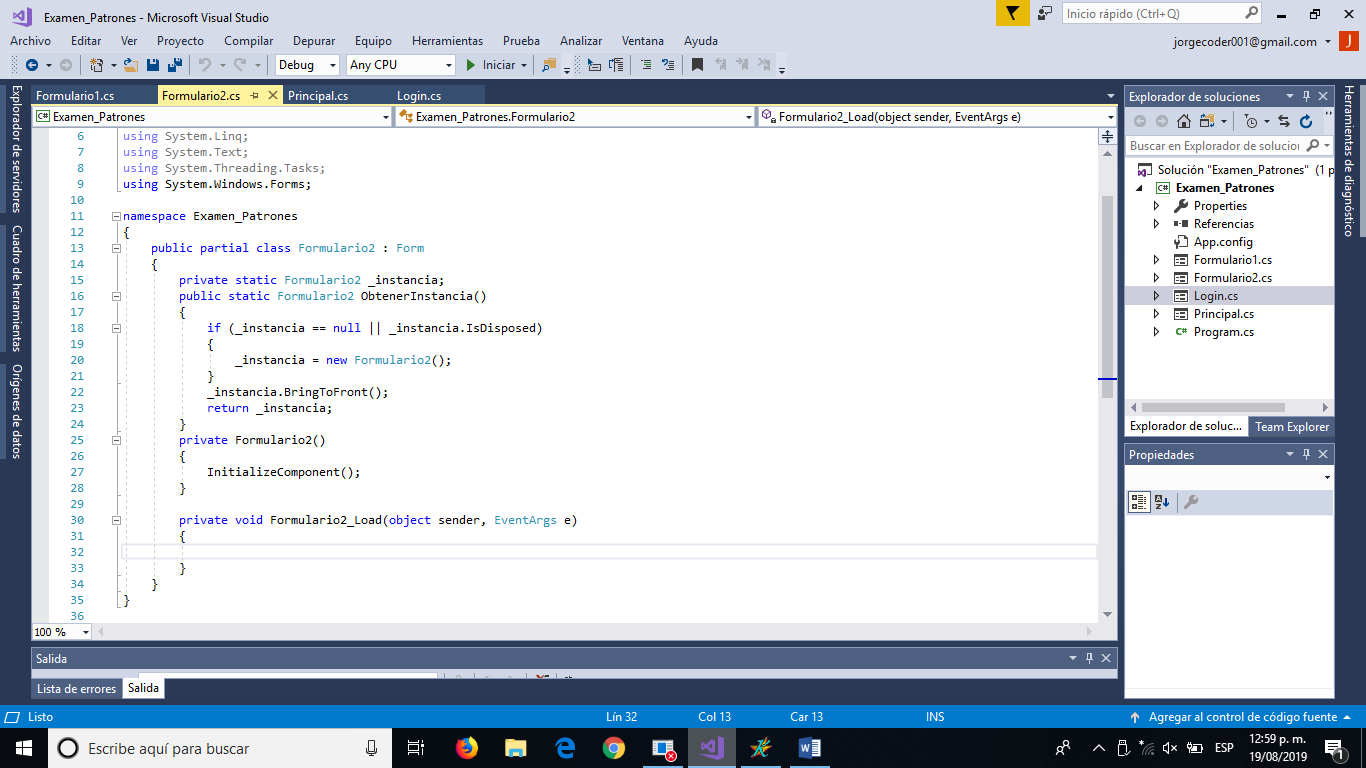


Figura Aplicación del patrón Singleton

En el formulario 2, se agregó el código anterior, el cual emplea el patrón de diseño Singleton, donde si el formulario 2 está abierto, al seleccionado nuevamente mostrará el que ya estaba iniciado en lugar de crear una ventana nueva a diferencia del formulario 1, que se va creando uno nuevo cada vez que se selecciona esa opción. Por lo tanto en el formulario 1, no se agregó ningún código diferente al que ya tenía por default.

# Funcionamiento

Al momento de ejecutar el programa, se muestran las ventanas siguientes, donde el usuario podrá iniciar sesión y la interfaz principal, la cual no se podrá modificar hasta que no se inicie sesión.

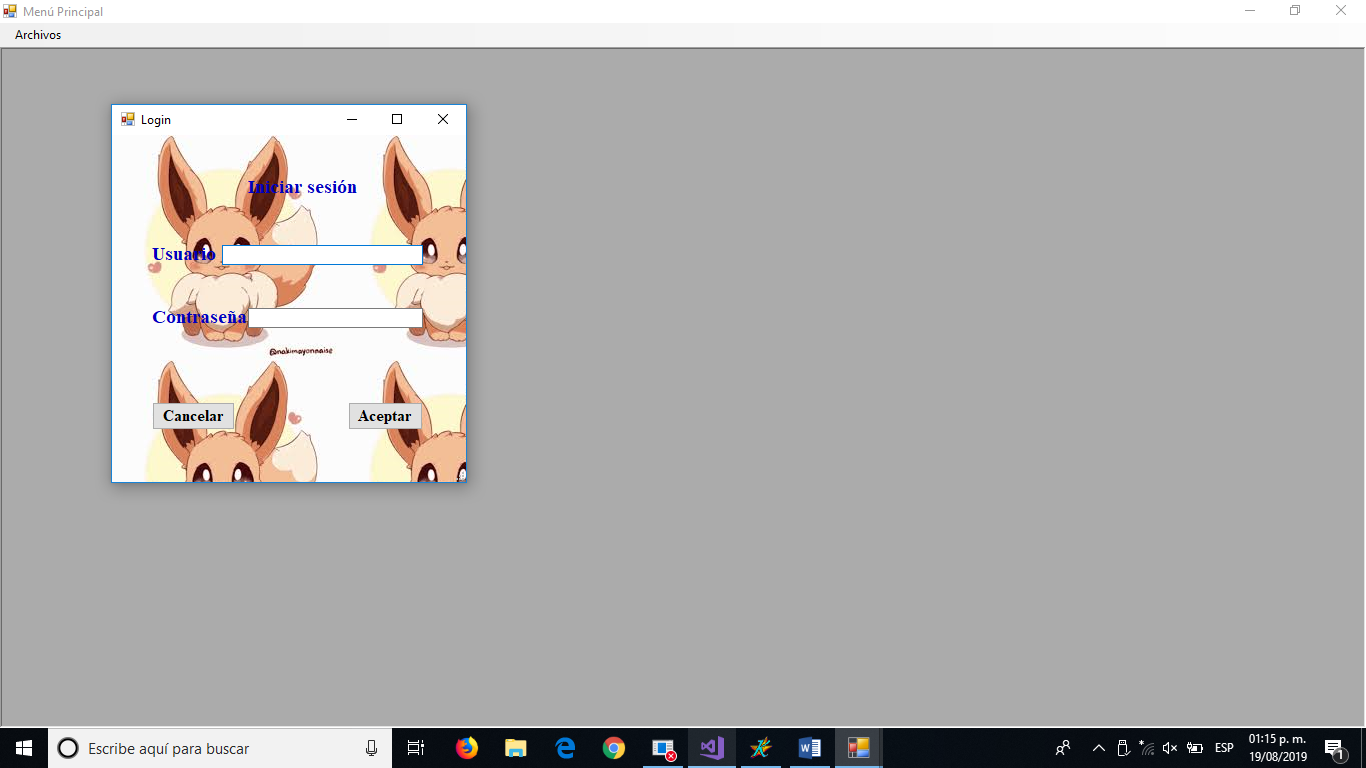


Figura Interfaz de login en ejecución

En caso de que algún dato de inicio de sesión, sea incorrecto, en la ventana, mostrará el mensaje dando a entender que algún dato está mal escrito, como se muestra en la imagen siguiente:

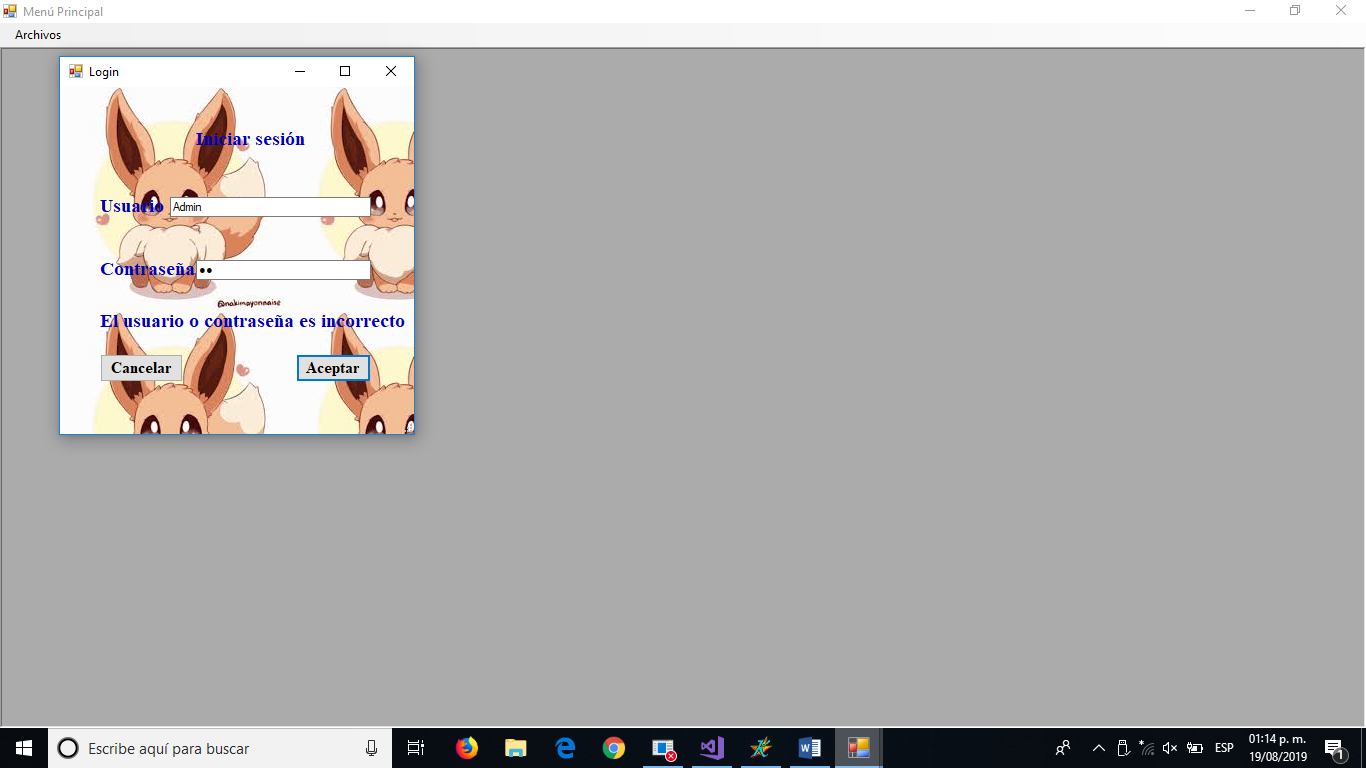


Figura Interfaz de login en ejecución con contraseña errónea

Una vez se inició la sesión, se muestra únicamente la ventana principal con el menú que dice archivos y del cual se despliegan los formularios 1 y 2 que son los que se ejecutará.

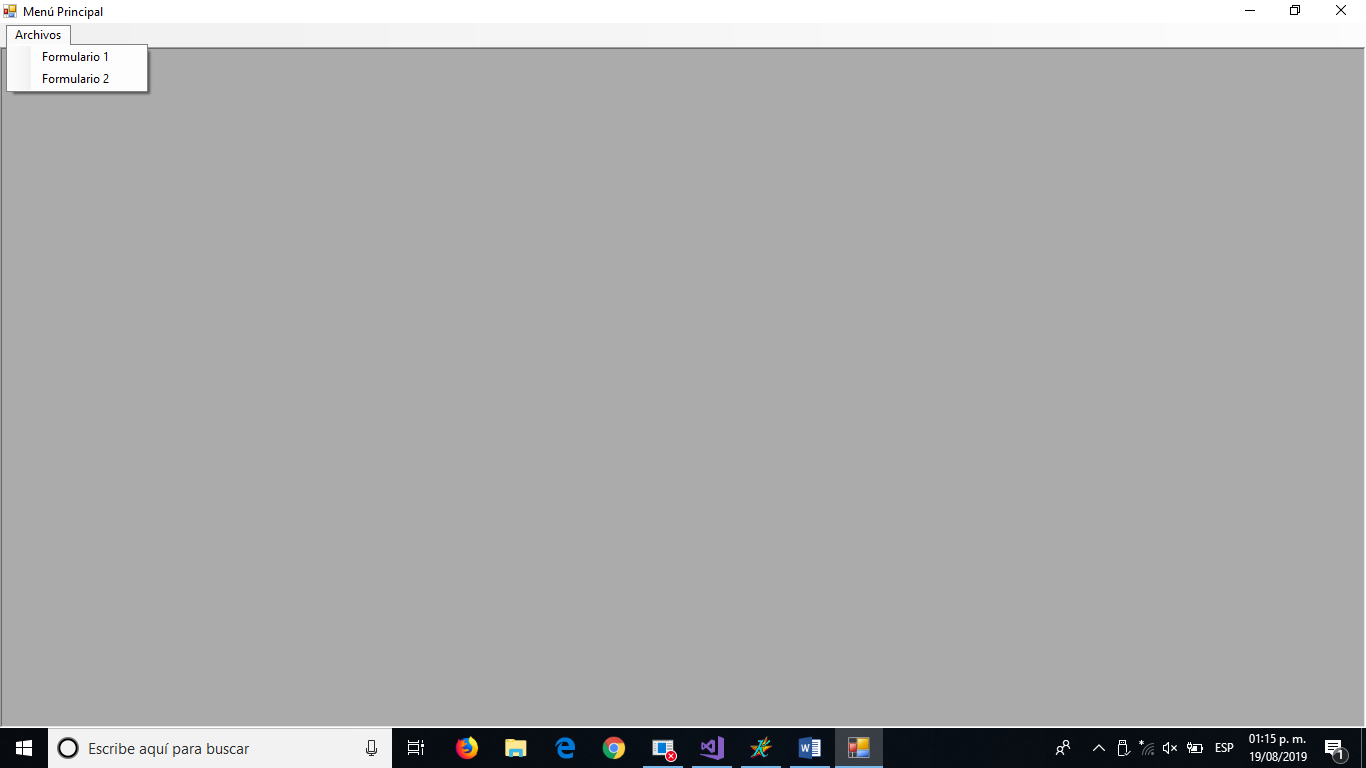


Figura Interfaz principal en ejecución para acceso a los formularios

Si se selecciona constantemente el formulario 1, crea nuevas instancias a este y por lo tanto, se crean muchas ventanas de este formulario, al cual se le colocó una imagen de fondo y varios textos, que lo diferenciarán del formulario 2, y al abrir muchas ventanas del formulario 1, se mostrará como la siguiente imagen:

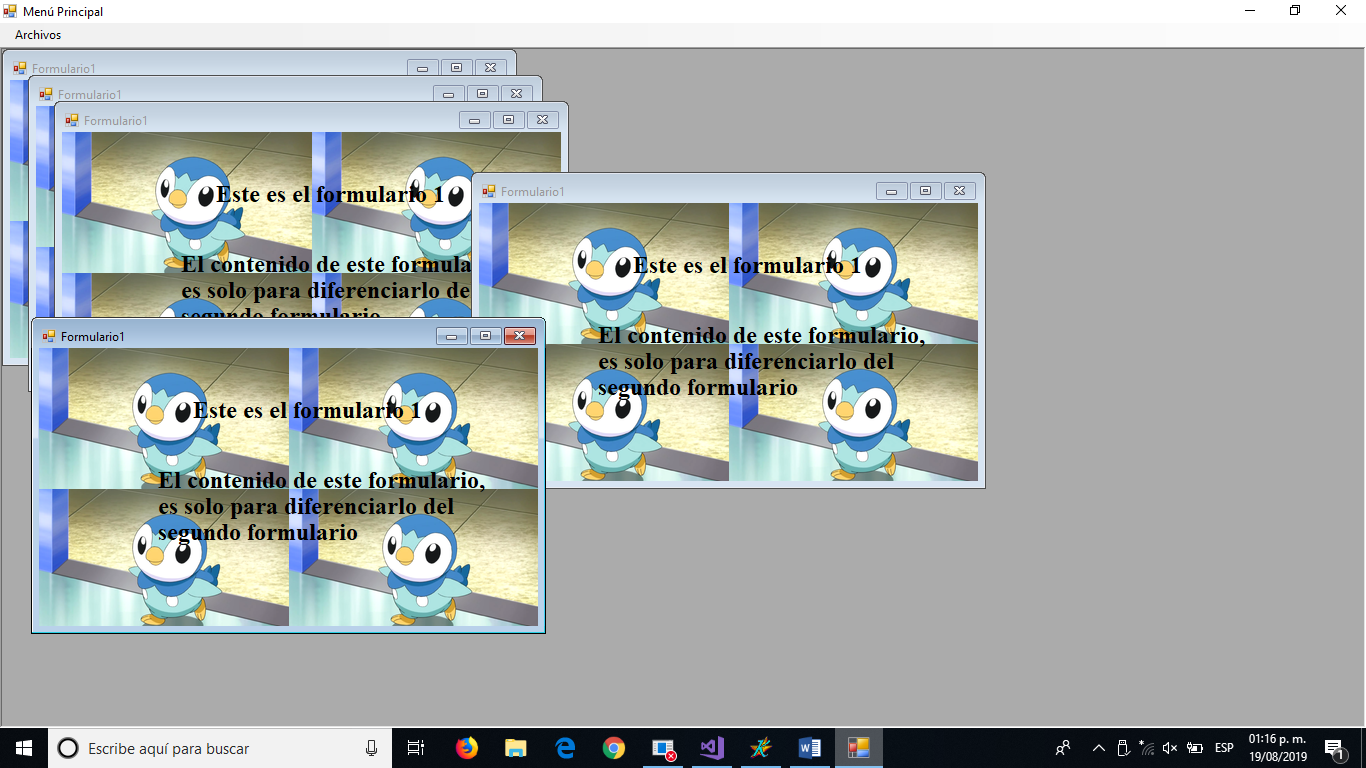


Figura Interfaz de formulario 1 en ejecución

Una vez ejecutamos el formulario 2, escribimos en la caja de texto para que se vea como en la imagen siguiente (se puede escribir lo que sea):

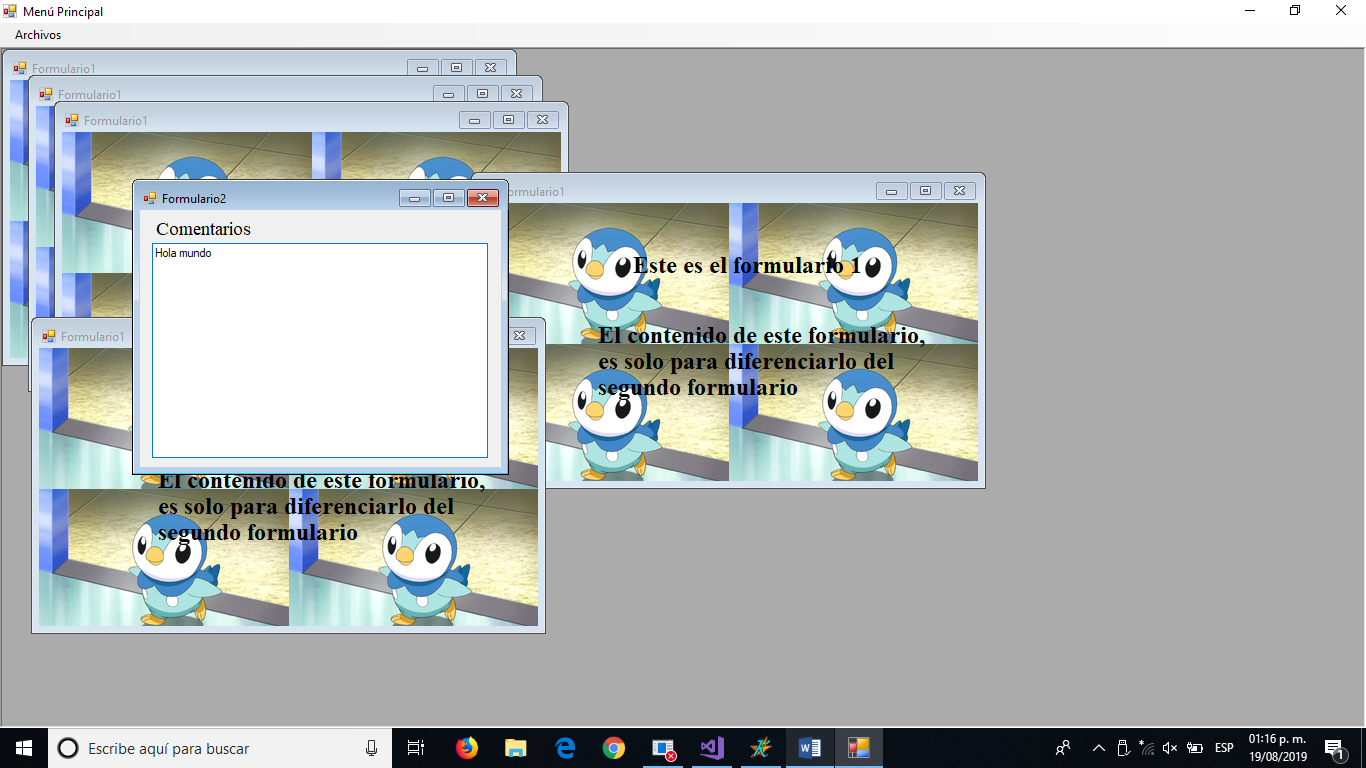


Figura Interfaz formulario 2 en ejecución

Después, se ejecutan nuevamente varias veces la opción de formulario 1, para abrir nuevas ventanas, dejando a la vista la ventana del formulario 2, tal y como se ven en la imagen siguiente:

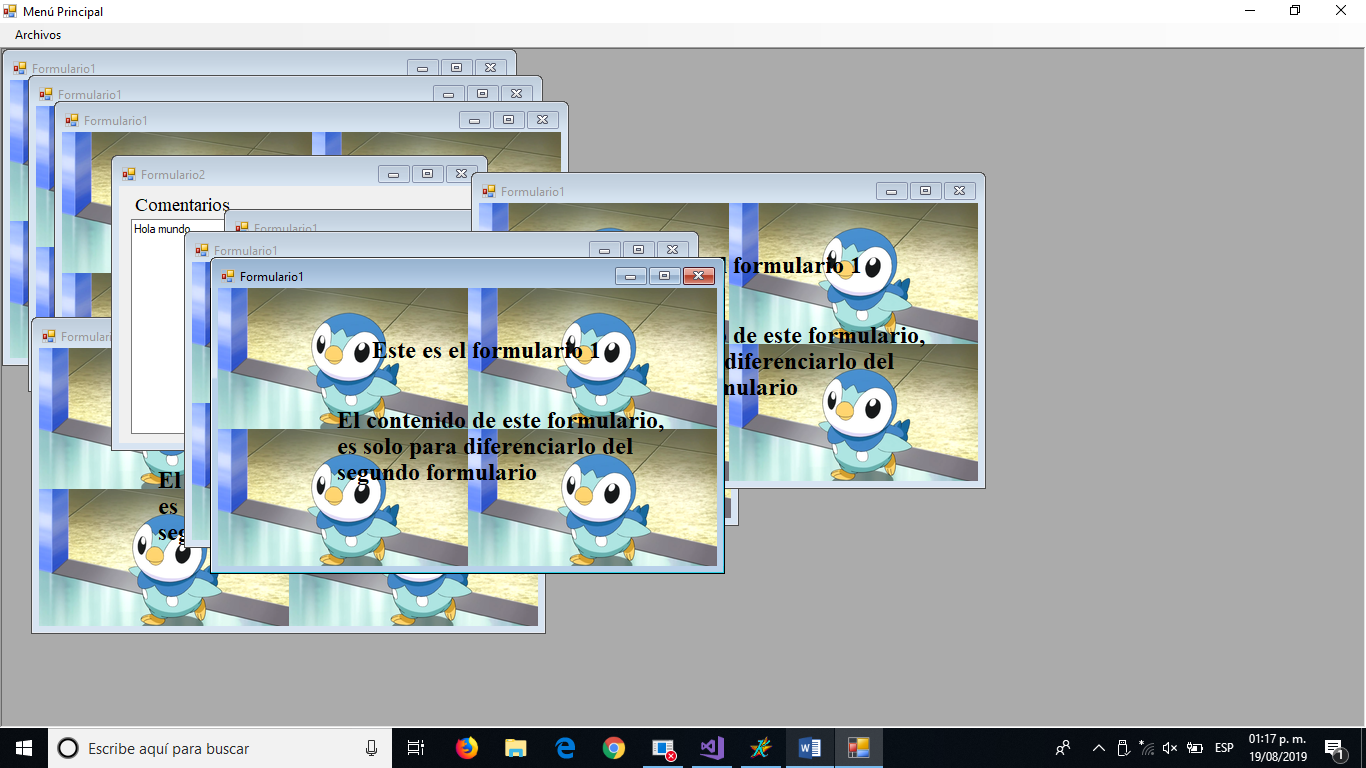


Figura Se vuelven a crear formularios 1

Y después, ejecutamos de nuevo la ventana de formulario 2, la cual, nos muestra la ventana que ya estaba siendo modificada, en lugar de crear una nueva, como se observa en la imagen siguiente:

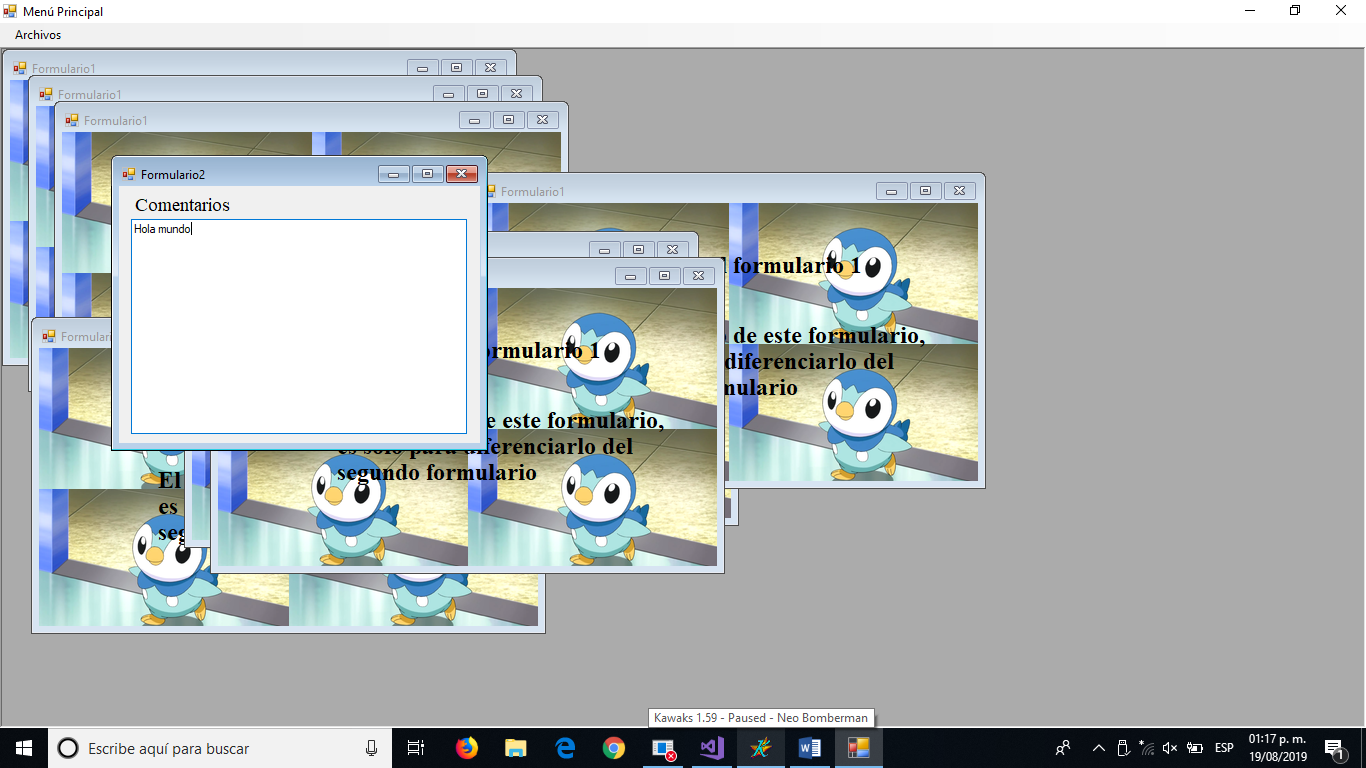


Figura Ejecución de patrón Singleton

Este comportamiento del formulario 2, ocurre gracias al uso del patrón de diseño Singleton, ya que al solo permitirnos instanciar una sola vez cada clase, como el formulario ya está creado (instanciado), nos mostrará ese formulario que ya está en lugar de crear uno nuevo.

Lo cual lo diferencia del formulario 1, ya que al no aplicar el patrón de diseño Singleton en este, cada que se seleccione el formulario 1, creará una nueva instancia de este y por lo tanto, abrirá una nueva ventana del formulario cada vez.