

Ofuscador de código fuente para proyectos realizados en .NET



Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos  
realizados en .NET

Duván Stiven Garzón Morales

Andrés Felipe Gutiérrez Ceballos

Corporación Universitaria Minuto de Dios

Rectoría Orinoquia

Sede / Centro Universitario Villavicencio (Meta)

Programa Tecnología en Desarrollo de Software

mayo de 2024

Ofuscador de código fuente para proyectos realizados en .NET

Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos  
realizados en .NET

Duván Stiven Garzón Morales

Andrés Felipe Gutiérrez Ceballos

Trabajo de Grado presentado como requisito para optar al título de Tecnólogo en Desarrollo de  
Software

Asesor 1:

Kelly Johanna Ávila Matías

Ingeniera Sistemas

Asesor 2:

Wilmer Hernández Álvarez

Ingeniero de Sistemas

Corporación Universitaria Minuto de Dios

Rectoría Orinoquia

Sede / Centro Tutorial Villavicencio (Meta)

Programa Tecnología en Desarrollo de Software

mayo de 2024

## Tabla de contenido

Lista de tablas .....	6
Lista de figuras.....	7
Lista de anexos.....	9
Resumen .....	10
Abstract .....	11
Introducción .....	12
CAPÍTULO I.....	13
1.1    Objetivo general .....	13
1.2    Objetivos específicos .....	13
1.3    Planteamiento del problema.....	13
1.3.1    Formulación del problema .....	14
1.4    Justificación.....	15
CAPITULO II.....	17
2.1    Marco teórico.....	17
2.2    Antecedentes teóricos .....	18
2.3    Marco conceptual .....	20
2.4    Marco legal.....	21
CAPITULO III.....	23
3.1    Tipo de investigación .....	23
3.2    Muestra .....	23

3.3	Instrumentos y técnicas de recolección de información .....	25
CAPITULO IV .....		26
4.1	Metodología de desarrollo de software .....	26
4.2	Análisis de requerimientos.....	26
4.2.1	Requisitos funcionales: .....	26
4.2.2	Requisitos no funcionales: .....	28
4.2.3	Requisitos de integración: .....	30
4.3	Historias de usuario .....	30
4.4	Diseño de la aplicación .....	31
4.4.1	Casos de uso.....	32
4.4.2	Diagrama de secuencia .....	33
4.4.3	Mockups .....	33
4.4.4	Diagrama de clases .....	36
4.5	Desarrollo del aplicativo .....	36
4.6	Diccionario de datos .....	38
4.7	Plan de pruebas .....	38
CAPITULO V.....		41
5.1	Análisis de datos .....	41
CAPITULO VI .....		49
6.1	Conclusiones .....	49

## Ofuscador de código fuente para proyectos realizados en .NET

6.2	Recomendaciones .....	50
6.3	Resumen Analítico Especializado – RAE .....	51
	Referencias .....	53
	Anexos.....	56
7.1	Inicio de sesión .....	56
7.2	Cargar Archivo y Ofuscar.....	57
7.3	Resultado final .....	58

**Lista de tablas**

Tabla 1: Requerimiento Funcional N°1 .....	26
Tabla 2: Requerimiento Funcional N°2.....	27
Tabla 3: Requerimiento Funcional N°3.....	27
Tabla 4: Requisito no funcional N°1 .....	28
Tabla 5: Requisito no funcional N°2 .....	28
Tabla 6: Requisito no funcional N°3 .....	29
Tabla 7: Requisito no funcional N°4 .....	29
Tabla 8. Historia de usuario N°1 .....	30
Tabla 9. Historia de usuario N°2.....	30
Tabla 10. Plan de pruebas .....	39
Tabla 11. Formato RAE.....	51

## Lista de figuras

Figura 1. Formula de Murray y Larry .....	24
Figura 2. Tamaño de la muestra.....	24
Figura 3. Casos de uso.....	32
Figura 4. Diagrama de secuencia .....	33
Figura 5. Mockup creación de seriales.....	34
Figura 6. Mockup de autenticación .....	35
Figura 7. Mockup inicio .....	35
Figura 8. Diagrama de clases.....	36
Figura 9. Fragmento de código C#.....	37
Figura 10. Diccionario de datos .....	38
Figura 11. Pregunta N°1 .....	41
Figura 12. Pregunta N°2 .....	43
Figura 13. Pregunta N°3 .....	43
Figura 14. Pregunta N°4 .....	44
Figura 15. Pregunta N°5 .....	45
Figura 16. Pregunta N°6 .....	46
Figura 17. Pregunta N°7 .....	47
<b>Figura 18.</b> Pregunta N°8 .....	47
<b>Figura 19.</b> Formulario de registro .....	56
Figura 20: Menú principal.....	57
Figura 21. Ofuscación .....	58

Figura 22. Vista previa archivo ofuscado ..... 59

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

### **Lista de anexos**

Anexo 1. Manual de usuario del Encriptador de archivos de desarrollo con herramientas .NET 56

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

### **Resumen**

El desarrollo de una aplicación de escritorio para la ofuscación de archivos fuente de proyectos realizados es una solución de seguridad para proteger código fuente de aplicaciones de escritorio en el ámbito del desarrollo de software. Este aplicativo utiliza el marco .NET de Microsoft, que proporciona una amplia gama de herramientas y librerías para implementar algoritmos criptográficos sólidos.

Una de las funcionalidades clave del aplicativo es el cifrado de código fuente, que permite a los desarrolladores proteger funciones importantes utilizadas durante el desarrollo de software. Esto asegura que solo los usuarios autorizados tengan acceso a los fragmentos vulnerables del código fuente y evitar posibles fugas de información que podrían poner en riesgo la propiedad intelectual o la privacidad, tanto para los desarrolladores independientes como para las empresas de desarrollo de software en la ciudad de Villavicencio.

**Palabras clave:** Ofuscación, .NET, Información confidencial, Desarrollo de software.

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

### **Abstract**

The development of an application for the obfuscation of source files of completed projects is a security solution to protect source code of desktop applications in the field of software development. This application uses Microsoft's .NET framework, which provides a wide range of tools and libraries to implement strong cryptographic algorithms.

One of the key functionalities of the application is source code encryption, which allows developers to protect important functions used during software development. This ensures that only authorized users have access to vulnerable fragments of the source code and avoids possible information leaks that could put intellectual property or privacy at risk, both for independent developers and software development companies in the city of Villavicencio.

**Keywords:** Obfuscation, .NET, Confidential information, Software development.

# **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

## **Introducción**

En la era digital actual, la protección de los datos sensibles y la seguridad de la información se han convertido en preocupaciones primordiales para las empresas de todos los sectores. En particular, las empresas de desarrollo de software enfrentan desafíos para garantizar la protección del código fuente de los proyectos, y a su vez, disminuir los riesgos de violaciones de seguridad y la necesidad de cumplir con las regulaciones de privacidad cada vez más estrictas.

Con el objetivo de abordar estas preocupaciones y proporcionar soluciones efectivas, se creó una aplicación de escritorio para empresas de desarrollo de software en Villavicencio, con la intención de salvaguardar y proteger el código fuente de los aplicativos desarrollados para los clientes.

La aplicación hace uso de técnicas de encapsulación, ofuscación individual y anti-desensamblador para hacer que el acceso, la modificación de código y la ingeniería inversa sean más desafiantes en caso de que un tercero lo intente. Estas técnicas añaden un nivel adicional de seguridad que va más allá del cifrado básico que se encuentra comúnmente en repositorios de GitHub.

Se espera que los resultados de este aplicativo contribuyan significativamente al campo de la seguridad en las aplicaciones web desarrolladas en la ciudad de Villavicencio.

## **CAPÍTULO I**

### **1.1 Objetivo general**

Desarrollar una herramienta para ofuscación de código fuente de proyectos de software dirigida a desarrolladores de Villavicencio.

### **1.2 Objetivos específicos**

Determinar si es viable el desarrollo de un sistema de protección de información mediante un ofuscador de archivos .NET.

Diseñar la arquitectura del aplicativo mediante wireframes creados a partir del levantamiento de requerimientos.

Desarrollar la aplicación mediante el uso de herramientas .NET y la implementación de algoritmos de cifrado y ofuscación.

### **1.3 Planteamiento del problema**

El problema principal abordado en este proyecto de investigación fue la vulnerabilidad del código fuente de proyectos desarrollados con .NET, que carece de medidas adecuadas de seguridad y encriptación. Esta falta de protección expone los archivos ejecutables (.EXE) y las bibliotecas de enlace dinámico (.dll) a riesgos significativos, incluyendo violaciones de privacidad, manipulación de datos y potenciales infracciones de propiedad intelectual.

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

La ausencia de encriptación en los archivos de código fuente representa un riesgo sustancial de fuga de información confidencial, manipulación de datos, piratería informática y violaciones de propiedad intelectual (Fluid Attacks, 2022). Estos problemas pueden acarrear consecuencias legales, daños a la reputación y pérdida de confianza por parte de los clientes.

En este sentido, el reciente ataque de ransomware a EPS Sanitas, reportado por El Tiempo, puede proporcionar un punto de partida para comprender las consecuencias de la falta de seguridad en sistemas informáticos (Alvarez, 2022). Además, la situación se ve reflejada en un informe de Enter.co, donde se detalla que Sanitas y Colsanitas continúan experimentando interrupciones en sus servicios digitales como consecuencia del ataque (Betancourt, 2022). Esta información refuerza la necesidad urgente de fortalecer las medidas de seguridad en entornos informáticos, donde la confidencialidad y la integridad de los datos son de vital importancia. En este contexto, el análisis de las deficiencias en la seguridad del código fuente de proyectos desarrollados con .NET cobra aún más relevancia, ya que su corrección podría contribuir significativamente a prevenir incidentes similares en el futuro.

Además, desde la perspectiva del cumplimiento normativo en Colombia, la protección de datos y la privacidad están reguladas por la Ley Estatutaria 1581 de 2012. Esta legislación establece disposiciones generales para el tratamiento de datos personales por parte de entidades públicas y privadas, supervisada por la Superintendencia de Industria y Comercio (SIC). Cumplir con esta normativa implica garantizar el respeto a los principios, derechos y procedimientos específicos para el manejo adecuado de la información personal en el país.

### **1.3.1 Formulación del problema**

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

¿De qué manera se puede desarrollar una aplicación para ofuscar código fuente de proyectos realizados en .NET por desarrolladores de software de la ciudad de Villavicencio?

### **1.4 Justificación**

La justificación de este proyecto se fundamenta en la necesidad crítica de proteger el código fuente de aplicaciones desarrolladas en entornos .NET frente a riesgos de seguridad y privacidad. Actualmente, la mayoría de las medidas de seguridad se centran en proteger los datos en tiempo de ejecución, mientras que el código fuente subyacente permanece vulnerable y expuesto.

La falta de encriptación y medidas de protección en el código fuente puede resultar en graves consecuencias, incluyendo la exposición de información confidencial a terceros no autorizados, la manipulación malintencionada de datos, y posibles infracciones de propiedad intelectual. Estos riesgos no solo amenazan la integridad y privacidad de los datos, sino que también pueden derivar en pérdidas financieras, daños reputacionales y repercusiones legales.

Este proyecto busca desarrollar una aplicación especializada en la ofuscación de código fuente para proyectos .NET, con el propósito de mitigar estas amenazas. La aplicación aprovechará técnicas de ofuscación para mejorar la seguridad y robustez del backend, ocultando la lógica interna y las estructuras del programa. Estas técnicas dificultarán la comprensión y manipulación del código por parte de usuarios no autorizados. La interfaz de usuario se mantendrá intuitiva y básica, ya que el enfoque principal del proyecto reside en el backend y en la solidez de sus funciones. Entre las medidas implementadas, se incluye la modificación del punto de entrada del programa para dificultar su análisis, la introducción de métodos de

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

ofuscación en el flujo de ejecución del código para complicar su trazabilidad, la optimización del rendimiento mediante la inserción directa del contenido de ciertos métodos en su lugar de llamada, la incorporación de métodos de descriptación según sea necesario para manejar la información cifrada de manera segura y eficiente, y la introducción de métodos falsos para confundir a posibles atacantes y dificultar la comprensión de la estructura y lógica del código. Estas estrategias trabajan en conjunto para reforzar la seguridad del aplicativo y proteger los datos sensibles que maneja.

Además, la implementación de medidas de encriptación del código fuente garantizará el cumplimiento con regulaciones de protección de datos, como la Ley Estatutaria 1581 de 2012 en Colombia. Esto proporcionará a las empresas y desarrolladores la seguridad necesaria para salvaguardar la información confidencial de acuerdo con los estándares legales y éticos vigentes.

En resumen, este proyecto persigue cubrir una brecha crítica en la seguridad de aplicaciones .NET al ofrecer una solución robusta y eficaz para la protección del código fuente, mitigando así los riesgos asociados con la exposición y manipulación no autorizada de datos sensibles.

## CAPITULO II

### 2.1 Marco teórico

El ofuscador de archivos de desarrollo con herramientas .NET aborda la problemática crucial de proteger la información sensible con tecnologías .NET. Su enfoque radica en la integración con el entorno, además de la utilización de algoritmos de encriptación modernos como AES (Microsoft, 2023), un manejo seguro de claves, soporte para diferentes tipos de datos, facilidad de integración en proyectos existentes y consideraciones de rendimiento para asegurar una protección sólida sin comprometer la eficiencia del sistema. Este aplicativo promete ser una herramienta valiosa para desarrolladores y empresas que busquen salvaguardar código fuente de forma efectiva, especialmente en un contexto donde las prácticas de codificación modernas se centran en la simplicidad del código compilado.

Estas prácticas incluyen el uso de lenguajes de alto nivel, diseño orientado a objetos, nombres significativos y patrones estándar, lo que conlleva a un código fuente más claro y comprensible. Sin embargo, esta claridad también puede facilitar la ingeniería inversa por parte de atacantes, quienes podrían aprovecharse para comprender el comportamiento del programa y descubrir protocolos de comunicación con sistemas back-end (Pipelines, 2023), subvirtiendo así los sistemas de defensa perimetral.

En el contexto actual, se destaca la importancia de implementar medidas de seguridad robustas en el desarrollo de software para proteger la propiedad intelectual y prevenir la distribución ilegal de productos. Adobe, una empresa líder en software de diseño y creatividad, ha implementado métodos como la ofuscación de código y la encriptación en sus aplicaciones emblemáticas como Photoshop, Illustrator y toda su suite. Estas medidas son esenciales para

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

garantizar la integridad de sus productos y la confianza de sus clientes en su autenticidad y calidad ( VMProtect Software, 2024).

En un contexto más amplio, la adopción de servicios de ofuscación es vital para combatir la piratería y la ingeniería inversa en la industria del software. Empresas como Ubisoft, reconocida por sus videojuegos, han recurrido a técnicas de protección como VMProtect en títulos como Assassin's Creed Origins. La combinación de VMProtect con otras protecciones estándar demostraron ser una estrategia efectiva, como se evidencia en el caso mencionado, donde el juego permaneció seguro durante un período crucial para las ventas (Maxwell, 2017).

La implementación de estas medidas no solo protege los intereses comerciales de las empresas, sino que también contribuye a mantener la integridad de la industria del software y a crear un ecosistema digital más seguro para todos los usuarios. Aunque muchas empresas mantienen en secreto los detalles específicos de sus estrategias de seguridad, la adopción de servicios de ofuscación y otras medidas de protección sigue siendo una práctica esencial en la lucha contra la piratería y la ingeniería inversa (VMProtect Software, 2023).

### **2.2 Antecedentes teóricos**

La protección de la información sensible en el mundo digital se ha convertido en una prioridad indiscutible en el ámbito de la ciberseguridad. En este contexto, es fundamental evaluar la eficacia de los productos de seguridad disponibles en el mercado para combatir las crecientes amenazas cibernéticas, especialmente en lo que respecta a la protección anti decompilación de código fuente.

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

Actualmente las empresas más notables para la ofuscación son YakPro, Eziriz y VMProtect. YakPro se destaca por su enfoque en técnicas de transformación de código, adaptándose a diversos lenguajes de programación, aunque su naturaleza de código abierto puede exponerlo a riesgos de decompilación (YAK Pro WPS, 2022). Por su parte, Eziriz ofrece soluciones más específicas para las necesidades de sus clientes .NET, con herramientas de ofuscación y encriptación utilizadas por grandes corporaciones como Intel, Microsoft y FedEx. Estas soluciones permiten una personalización detallada de la protección, virtualización de código desde niveles básicos hasta avanzados con defensas contra depuración y análisis de memoria (.NET Reactor Eziriz, 2023). En cuanto a VMProtect, es la herramienta más confiable actualmente para proteger aplicaciones y juegos contra la ingeniería inversa y manipulaciones no autorizadas. Empresas de renombre como Adobe, Ubisoft, Electronic Arts y Activision han integrado VMProtect en sus productos. Su característica distintiva radica en su capacidad para implementar técnicas de encriptación en datos críticos, ofreciendo una defensa robusta y multifacética. A diferencia de otras herramientas, proporciona una variedad de opciones de protección, desde la ofuscación hasta la generación de códigos de máquina virtuales, lo que dificulta la comprensión del funcionamiento interno del software (VMProtect Software, 2023). Sin embargo, el uso de estos ofuscadores a la medida tiende a tener costos excesivamente altos, lo que no retribuye directamente al mercado colombiano, En este contexto, la evaluación de la eficacia de estas soluciones se vuelve crucial para garantizar una protección sólida frente a las amenazas cibernéticas en constante evolución.

(Kaspersky Corporation, 2023) llevó a cabo un estudio con el propósito de analizar la eficacia de sus productos de seguridad para detectar y prevenir ataques anti decompilación en diferentes segmentos, incluyendo consumidores finales, pequeñas empresas y clientes

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

independientes. Se evaluó la eficacia de los productos de seguridad, específicamente en el ámbito de programas maliciosos, aunque se identificó un porcentaje de ineficiencia en el sector de ofuscación. Este hallazgo sugiere que dichos productos abarcan múltiples aspectos, sin garantizar una protección directamente proporcional al código fuente del aplicativo, lo que conlleva la presencia de pequeñas vulnerabilidades.

### **2.3 Marco conceptual**

Para la implementación del aplicativo, se optó por aprovechar las librerías y herramientas ofrecidas por la plataforma .NET, en particular, se hizo uso de *System.Security.Cryptography* y *Entity Framework*. La selección de *System.Security.Cryptography* se fundamenta en su capacidad para aplicar algoritmos de encriptación robustos, garantizando así la protección adecuada de archivos y datos sensibles. Esta biblioteca proporciona una amplia gama de algoritmos criptográficos confiables que pueden adaptarse a los requisitos específicos de seguridad del sistema (Microsoft Corporation, 2022).

Por otro lado, la elección de *Entity Framework* se justifica por su capacidad para simplificar la interacción con la base de datos, permitiendo un manejo eficiente y seguro de los datos cifrados. *Entity Framework* facilitó la implementación de capas de acceso a datos y Mapeo Objeto-Relacional (ORM), lo que redujo la complejidad del desarrollo y mejora la mantenibilidad del código. Además, al integrarse con *System.Security.Cryptography*, se garantizó un almacenamiento seguro de la información en la base de datos, cumpliendo con los estándares de seguridad y privacidad requeridos por el aplicativo (Microsoft Corporation, 2022).

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

Además, se introdujeron diversas técnicas de protección y ofuscación. La función *ObfuscateEntryPoint* se encarga de modificar el punto de entrada del programa, alterando su estructura subyacente para dificultar la comprensión de su funcionamiento por parte de posibles atacantes. *AddFlowObfuscation* implementa métodos de ofuscación en el flujo de ejecución del código, complicando aún más la trazabilidad de las instrucciones y aumentando la resistencia frente a análisis de ingeniería inversa. *InlineMethodIfNeeded* se utiliza para optimizar la eficiencia del programa al incrustar el contenido de ciertos métodos en su lugar de llamada, reduciendo así la carga de llamadas a funciones y mejorando el rendimiento general del sistema. Por otro lado, *AddDecryptMethodIfNeeded* se emplea para agregar métodos de descryptación según sea necesario, asegurando que el sistema esté preparado para manejar la información cifrada de manera segura y eficiente (Code Maze , 2023). Finalmente, *AddFakeMethods* introduce métodos falsos que confunden a los atacantes, añadiendo ruido al análisis y dificultando la comprensión de la estructura y lógica del código. Estas implementaciones aumentan la seguridad del aplicativo frente a amenazas (cecilphillip, 2013).

### **2.4 Marco legal**

Ley 1266 de 2008 establece disposiciones generales sobre el hábeas data y regula el manejo de información en bases de datos personales, su objetivo principal es proteger los derechos fundamentales de las personas en relación con su información personal, garantizando privacidad y seguridad (Congreso De La Republica De Colombia, 2008).

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

La Ley 527 de 1999 regula el acceso y uso de datos, el comercio electrónico y las firmas digitales, estableciendo requisitos para su validez jurídica y responsabilidades de las entidades certificadoras (Congreso De La Republica De Colombia, 1999).

## **CAPITULO III**

### **3.1 Tipo de investigación**

El tipo de investigación seleccionado para este proyecto es la investigación aplicada con un diseño cuantitativo, alineada con las líneas de investigación definidas por UNIMINUTO. Se llevó a cabo con el respaldo del semillero Movilsoft y el grupo de investigación GITSAI. Esta elección se justifica debido a que el objetivo principal del proyecto es generar un producto que tiene aplicaciones prácticas directas en el campo de la programación y la seguridad de software. La investigación aplicada se centra en la resolución de problemas prácticos y la aplicación de los resultados de la investigación en la vida real, que es la finalidad de este proyecto.

Además, la investigación y desarrollo de este proyecto pueden abrir nuevas vías para futuros avances en la línea de investigación "Innovaciones Sociales y Productivas", por lo tanto, la selección de la Investigación Aplicada como el tipo de investigación para este proyecto se basa en su relevancia para el campo de estudio, su aplicabilidad práctica y su potencial para impulsar innovaciones en la seguridad del software y la programación

### **3.2 Muestra**

El tamaño ideal de la muestra se obtuvo siguiendo la fórmula propuesta por Murray y Larry (2005):

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

**Figura 1.** Formula de Murray y Larry

$$n = \frac{Z^2 \sigma^2 N}{e^2(N-1) + Z^2 \sigma^2}$$

Fuente: Libro Metodología de la investigación (2019)

Se muestra en la figura 1 los elementos y las operaciones correspondientes a realizar para calcular la muestra, en donde:

n = Es el tamaño de la muestra poblacional a obtener

N = Es el tamaño de la población total.

$\sigma$  representa la desviación estándar de la población, cuyo valor utilizado fue una constante que equivale a 0.5.

Z = Es el valor obtenido mediante niveles de confianza, cuyo valor es de 95% (1.96)

e = Representa el límite aceptable de error muestral, siendo este del 5%, el valor estándar usado en las investigaciones.

**Figura 2.** Tamaño de la muestra

Calcula el tamaño de tu muestra

Tamaño de la población (%) 58

Nivel de confianza (%) 95

Margen de error (%) 5

Tamaño de la muestra

**51**

Fuente: <https://es.surveymonkey.com/mp/sample-size-calculator/>

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

Mediante el uso de una herramienta virtual, se introdujeron los datos pertinentes para calcular el tamaño de muestra necesario. Con una población de 58, la herramienta arrojó un tamaño de muestra de 51, sugiriendo que un estudio significativo podría realizarse con menos individuos de lo esperado inicialmente.

### **3.3 Instrumentos y técnicas de recolección de información**

En el marco del proyecto, se emplearon instrumentos y técnicas de recolección de información con el propósito de obtener datos relevantes sobre la percepción y comprensión de la ofuscación de código en el contexto de la seguridad de software. Para lograr esto, se realizó una encuesta a un total de 40 desarrolladores independientes y 11 empresas de desarrollo de software de programación en la ciudad de Villavicencio. El enfoque utilizado incluyó preguntas cerradas de opción múltiple para evaluar el nivel de conocimiento y necesidad del producto.

## **CAPITULO IV**

### **4.1 Metodología de desarrollo de software**

El desarrollo del proyecto se ejecutó bajo la metodología ágil de desarrollo de software Extreme Programming (XP), aprovechando su flexibilidad para adaptarse a las necesidades cambiantes. Este enfoque iterativo e incremental permitió la entrega temprana de funcionalidades y ajustes continuos, asegurando así una respuesta ágil a los requisitos emergentes.

### **4.2 Análisis de requerimientos**

Se completó una evaluación de los requisitos necesarios para desarrollar un ofuscador de código para archivos .exe y .dll en el entorno .NET. Este análisis se centró en definir con precisión y detalle los objetivos y requerimientos del proyecto, proporcionando así una base sólida para todo el proceso de desarrollo y asegurando que el resultado final cumpla con las expectativas establecidas.

#### **4.2.1 Requisitos funcionales:**

**Tabla 1:** Requerimiento Funcional N°1

<b>Identificación del requerimiento:</b>	01
<b>Nombre del Requerimiento:</b>	Ofuscación de código

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

<b>Características:</b>	El sistema debe ser capaz de ofuscar el código fuente de archivos .exe y .dll desarrollados en el entorno .NET, alterando su estructura y lógica interna para dificultar la ingeniería inversa.
<b>Descripción del requerimiento:</b>	Se requiere que sea fácil de integrar en el flujo de desarrollo y configurable según las necesidades del proyecto.
<b>Prioridad del requerimiento:</b>	
Alta	

Fuente: elaboración propia (2024)

**Tabla 2:** Requerimiento Funcional N°2

<b>Identificación del requerimiento:</b>	02
<b>Nombre del Requerimiento:</b>	Preservación de funcionalidad
<b>Características:</b>	El proceso de ofuscación no debe comprometer la funcionalidad original del archivo .exe o .dll. Debe asegurarse de que las funciones y comportamientos del programa sigan siendo coherentes después de la ofuscación.
<b>Descripción del requerimiento:</b>	Es crucial que el aplicativo de escritorio mantenga sus funciones y comportamientos intactos después de aplicar la ofuscación.
<b>Prioridad del requerimiento:</b>	
Alta	

Fuente: elaboración propia (2024)

**Tabla 3:** Requerimiento Funcional N°3

<b>Identificación del requerimiento:</b>	03
<b>Nombre del Requerimiento:</b>	Soporte de plataforma
<b>Características:</b>	El ofuscador debe ser compatible con múltiples versiones de la plataforma .NET, asegurando que pueda utilizarse en diversos entornos de desarrollo y sistemas operativos.

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

<b>Descripción del requerimiento:</b>	Cumplimiento de licencias de terceros y normativas de seguridad.
<b>Prioridad del requerimiento:</b>	Alta

Fuente: elaboración propia (2024)

### 4.2.2 Requisitos no funcionales:

**Tabla 4:** Requisito no funcional N°1

<b>Identificación del requerimiento:</b>	N01
<b>Nombre del Requerimiento:</b>	Interfaz de usuario
<b>Características:</b>	El aplicativo de escritorio presentará una interfaz de usuario sencilla para que sea de fácil manejo para todo el que tenga acceso a él.
<b>Descripción del requerimiento:</b>	El aplicativo debe tener una interfaz de uso intuitiva y sencilla.
<b>Prioridad del requerimiento:</b>	Alta

Fuente: elaboración propia (2024)

**Tabla 5:** Requisito no funcional N°2

<b>Identificación del requerimiento:</b>	N02
<b>Nombre del Requerimiento:</b>	Rendimiento
<b>Características:</b>	El proceso de ofuscación será rápido y eficiente, minimizando el impacto en los tiempos de compilación y ejecución del programa.

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

<b>Descripción del requerimiento:</b>	Debe mantener un equilibrio entre la seguridad proporcionada por la ofuscación y el rendimiento general del sistema.
<b>Prioridad del requerimiento:</b>	
Alta	

Fuente: elaboración propia (2024)

**Tabla 6:** Requisito no funcional N°3

<b>Identificación del requerimiento:</b>	N03
<b>Nombre del Requerimiento:</b>	Seguridad
<b>Características:</b>	El ofuscador debe implementar técnicas sólidas de seguridad para proteger el código fuente contra la ingeniería inversa.
<b>Descripción del requerimiento:</b>	Debe resistir los intentos de decompilación y análisis estáticos para garantizar la protección de la propiedad intelectual del desarrollador
<b>Prioridad del requerimiento:</b>	
Alta	

Fuente: elaboración propia (2024)

**Tabla 7:** Requisito no funcional N°4

<b>Identificación del requerimiento:</b>	N04
<b>Nombre del Requerimiento:</b>	Facilidad de Uso
<b>Características:</b>	El sistema será intuitivo y fácil de usar, incluso para usuarios sin experiencia técnica avanzada.
<b>Descripción del requerimiento:</b>	Proporcionará interfaces gráficas y documentación clara para guiar a los usuarios a través del proceso de ofuscación de manera efectiva.
<b>Prioridad del requerimiento:</b>	
Alta	

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

Fuente: elaboración propia (2024)

### 4.2.3 Requisitos de integración:

**Integración de compartir en redes sociales:** Agregar botones para que los usuarios puedan compartir fácilmente contenido del sitio en sus redes sociales.

**Integración de Feed:** Mostrar las publicaciones más recientes de las cuentas sociales del cliente en el sitio web.

### 4.3 Historias de usuario

**Tabla 8.** Historia de usuario N°1

<b>Número:</b> 1	<b>Usuario:</b> Usuario
<b>Nombre historia:</b> Registro	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> 1
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador Responsable:</b> Duván S. Garzón - Andrés F. Gutiérrez	
<b>Descripción:</b> Quiero tener un registro único y no tener que estar validando mis credenciales.	
<b>Validación:</b> El acceso al inicio del aplicativo requiere que el usuario disponga de un código de ingreso único (serial) que garantice la seguridad y autenticación del acceso.	

Fuente: elaboración propia (2024)

**Tabla 9.** Historia de usuario N°2

<b>Número:</b> 2	<b>Usuario:</b> Usuario
------------------	-------------------------

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

<b>Nombre historia:</b> Home	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> 2
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 1
<b>Programador Responsable:</b> Duván S. Garzón - Andrés F. Gutiérrez	
<b>Descripción:</b> Quiero poder interactuar fácilmente con el aplicativo tanto con la sección de la ruta del archivo (.dll o .exe) como una vista previa de los nuevos módulos creados por la aplicación.	
<b>Validación:</b> Cuando el usuario hace clic en el botón “Cargar Archivo”, puede seleccionar el archivo deseado, y automáticamente la ruta de dicho archivo se muestra en la parte superior. Posteriormente, durante el proceso de ofuscación, el aplicativo proporciona una vista previa de los módulos generados para una mejor comprensión y validación.	

Fuente: elaboración propia (2024)

### 4.4 Diseño de la aplicación

La interfaz móvil del aplicativo se centra en la simplicidad y la seguridad, comenzando con un intuitivo apartado de registro para el ingreso del serial. Este proceso, esencial para activar las funciones del backend, es el primer paso hacia una protección robusta de los archivos.

Una vez registrado, el usuario accede al núcleo del aplicativo: el backend. Aquí, se despliega un proceso meticuloso de encapsulamiento y creación de módulos, diseñado para ofuscar el aplicativo sin comprometer su funcionalidad. La arquitectura de este sistema se ha concebido cuidadosamente para garantizar que el proceso de ofuscación se realice de manera

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

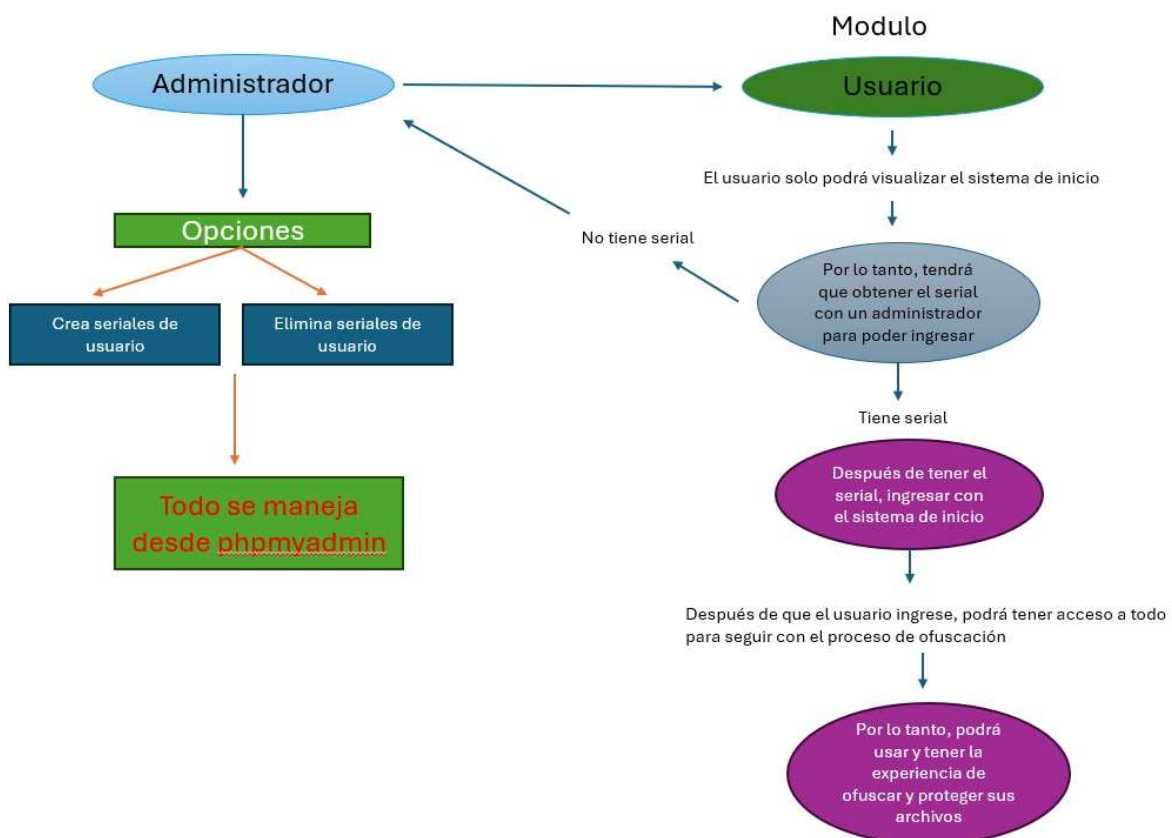
eficiente y sin incidir en el arranque o rendimiento del software. Cada aspecto del diseño está orientado a brindar una experiencia fluida y segura.

Con un enfoque en el diseño, la aplicación móvil ofrece una solución completa para mejorar la seguridad de los archivos .dll y .exe en entornos de trabajo .NET.

### 4.4.1 Casos de uso

Una vez establecidos los factores de registro, los usuarios interactuarán con el aplicativo (Figura 3). Este proceso constituye la interacción entre el usuario y la aplicación.

**Figura 3.** Casos de uso



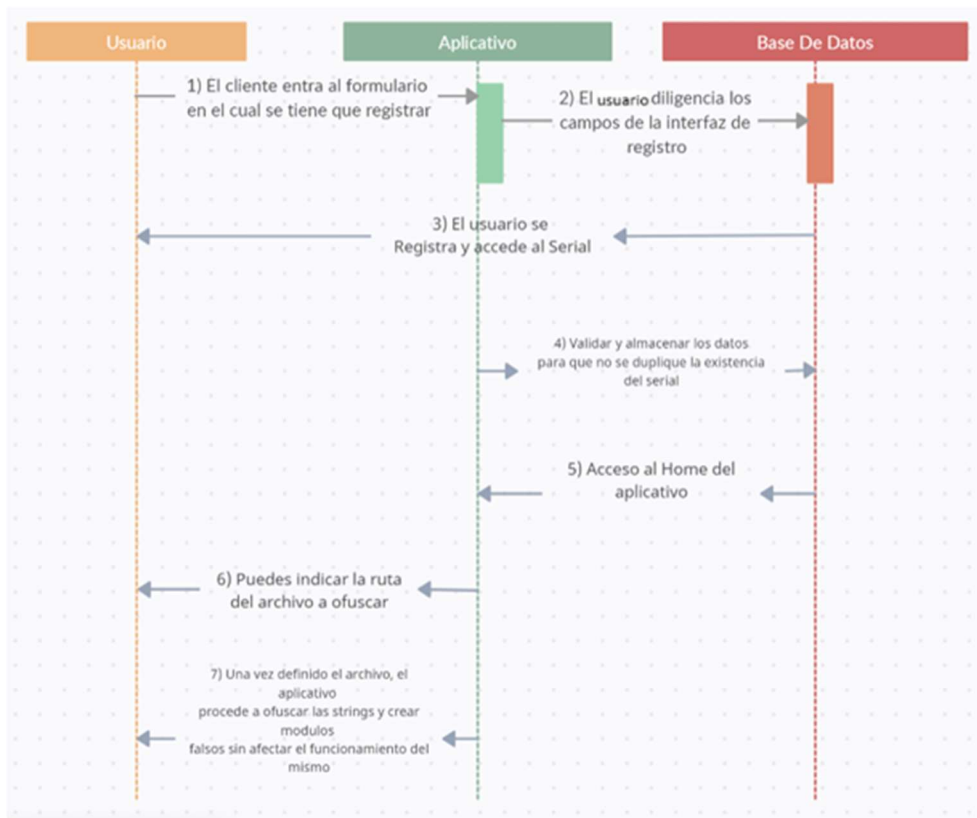
Fuente: elaboración propia (2024)

# Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

## 4.4.2 Diagrama de secuencia

Una vez planteado el diseño de los casos de uso con ayuda de la herramienta Creately se crea el diagrama de secuencia (Figura 4), en el cual se establecen los procesos que corresponden a una serie de acciones realizadas por el usuario.

Figura 4. Diagrama de secuencia



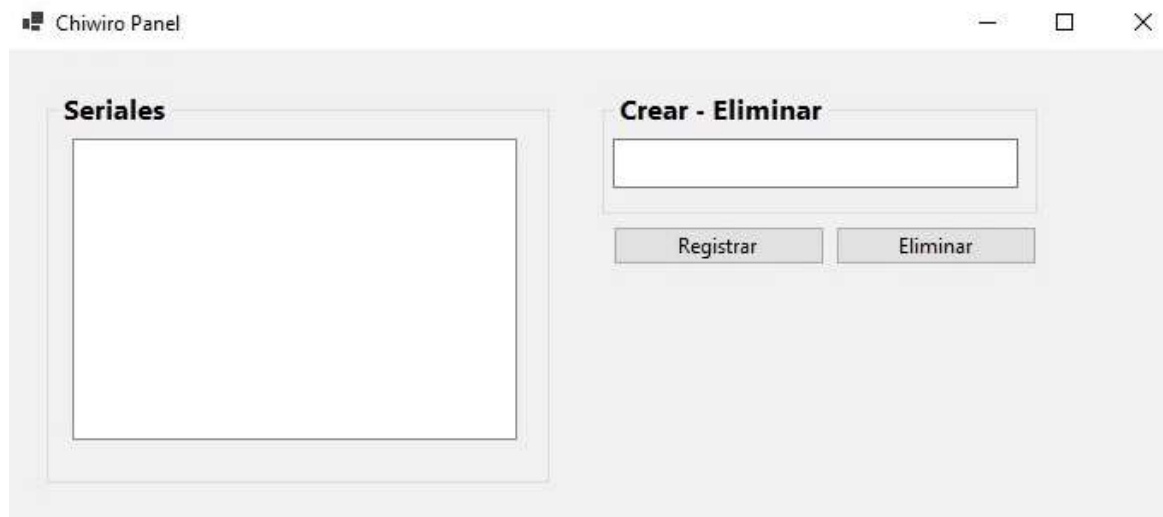
Fuente: elaboración propia (2024)

## 4.4.3 Mockups

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

Una vez establecidos los procesos, se diseñó una interfaz sencilla e intuitiva para los usuarios. Inicialmente se diseñó un apartado de implementación de seriales los cuales una vez registrados en el aplicativo tendrán una vigencia de 30 días a partir de la fecha (Figura 5).

**Figura 5.** Mockup creación de seriales



Fuente: elaboración propia (2024)

El módulo de autenticación solicita al usuario un código con vigencia de 30 días. Este código será suministrado por nuestro equipo de desarrollo y debe ser ingresado en el campo correspondiente (Figura 6).

# Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

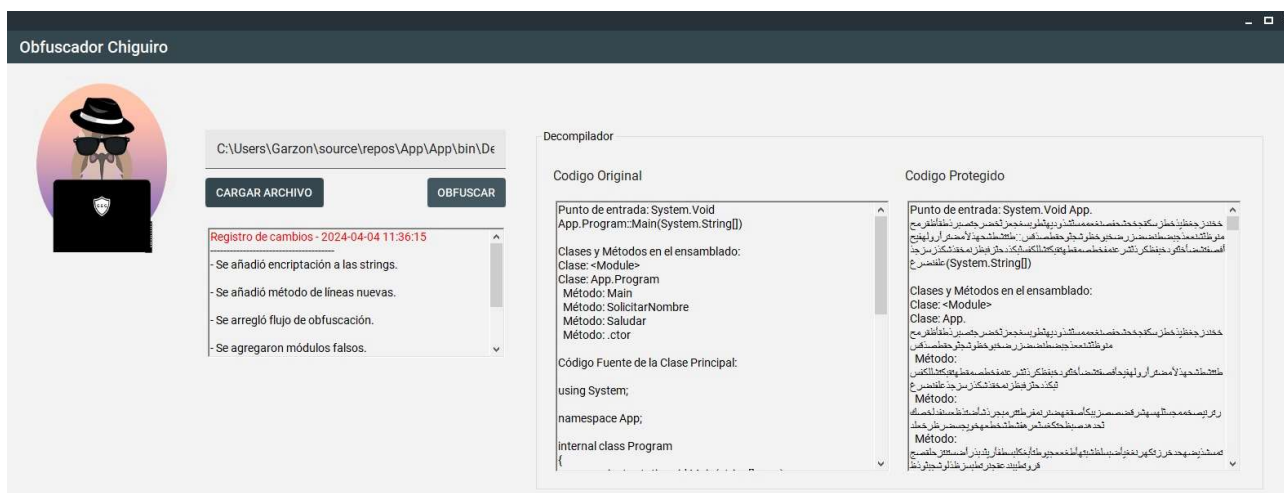
Figura 6. Mockup de autenticación



Fuente: elaboración propia (2024).

Después de registrar el serial, el aplicativo redirigirá al siguiente apartado (Figura 7). Se presentan dos botones: uno destinado a cargar el archivo a ofuscar y otro para ejecutar la ofuscación del archivo .dll o .exe. En la parte lateral de la interfaz, se muestra un preview de los nuevos módulos falsos y las strings alteradas.

Figura 7. Mockup inicio



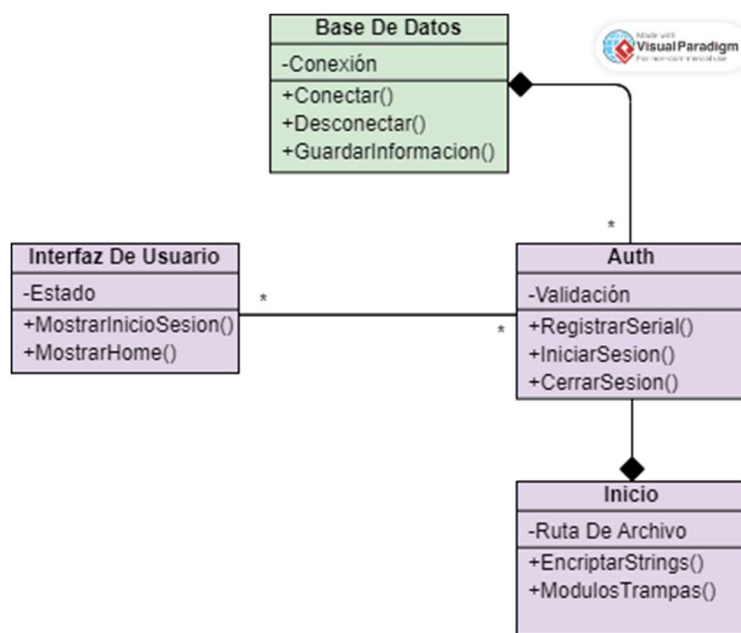
Fuente: elaboración propia (2024).

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

### 4.4.4 Diagrama de clases

Una vez definidas las funciones y los procesos, se procede a elaborar una representación visual (Figura 8) de los posibles elementos que harán parte del código, junto a sus atributos correspondientes, con el objetivo de asegurar un rendimiento óptimo.

**Figura 8.** Diagrama de clases



Fuente: elaboración propia (2024).

### 4.5 Desarrollo del aplicativo

El desarrollo del ofuscador se llevó a cabo en el entorno de desarrollo de C# dentro del ampliamente utilizado framework .NET (Figura 9). Para la creación de una interfaz de usuario atractiva, se optó por integrar la librería MaterialSkin.dll, reconocida por su estilo de diseño

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

limpio y contemporáneo, En cuanto a la ofuscación, se implementó una capa adicional de protección mediante la encriptación de módulos y cadenas utilizando la librería dnlib.dll. Esta herramienta no solo asegura la confidencialidad de los archivos .dll o .exe, sino que también añade una capa de complejidad que dificulta el análisis y la ingeniería inversa.

Finalmente, se estableció una sólida conexión con la base de datos mediante la inclusión de MySql.Data.dll, lo que garantiza un manejo eficiente y seguro de los datos.

Figura 9. Fragmento de código C#

```
private static void AddDecryptMethodIfNeeded(ModuleDef module)
{
    TypeDef stringDecryptorTypeDef = module.Find("StringDecryptor", false);
    if (stringDecryptorTypeDef == null)
    {
        stringDecryptorTypeDef = new TypeDefFusac("", "StringDecryptor", module.CorLibTypes.Object.TypeDefOrRef);
        module.Types.Add(stringDecryptorTypeDef);
        stringDecryptorTypeDef.Attributes = TypeAttributes.Public | TypeAttributes.Abstract | TypeAttributes.Sealed;

        MethodDef decryptMethod = new MethodDefFusac(
            "Decrypt",
            MethodSig.CreateStatic(module.CorLibTypes.String, module.CorLibTypes.String),
            MethodAttributes.Public | MethodAttributes.Static);
        stringDecryptorTypeDef.Methods.Add(decryptMethod);

        CilBody il = new CilBody();
        decryptMethod.Body = il;
        il.MaxStack = 2;

        il.Instructions.Add(OpCodes.Ldarg_0.ToInstruction());
        il.Instructions.Add(OpCodes.Call.ToInstruction(module.Import(typeof(Convert).GetMethod("FromBase64String", new[] { typeof(string) })))));
        Local localVar = new Local(module.Import(typeof(byte[])));
        decryptMethod.Body.Variables.Add(localVar);
        il.Instructions.Add(OpCodes.Stloc.ToInstruction(localVar));

        il.Instructions.Add(OpCodes.Call.ToInstruction(module.Import(typeof(Encoding).GetProperty("UTF8").GetMethod)));
        il.Instructions.Add(OpCodes.Ldloc.ToInstruction(localVar));
        il.Instructions.Add(OpCodes.Callvirt.ToInstruction(module.Import(typeof(Encoding).GetMethod("GetString", new[] { typeof(byte[]) })))));
        il.Instructions.Add(OpCodes.Ret.ToInstruction());
    }
}

EncryptStringsInModule(module, stringDecryptorTypeDef.Methods.First(m => m.Name == "Decrypt"));

private static void EncryptStringsInModule(ModuleDef module, MethodDef decryptMethod)
{
    foreach (var type in module.Types)
    {
        foreach (var method in type.Methods)
        {
            if (method.Body == null) continue;

            for (int i = 0; i < method.Body.Instructions.Count; i++)
            {
                Instruction instr = method.Body.Instructions[i];
                if (instr.OpCode == OpCodes.Ldstr)
                {
                    string encryptedString = EncryptString(instr.Operand as string);
                    instr.Operand = encryptedString;
                    Instruction callDecrypt = Instruction.Create(OpCodes.Call, decryptMethod);
                    method.Body.Instructions.Insert(i + 1, callDecrypt);
                    i++;
                }
            }
        }
    }
}
```

Fuente: elaboración propia (2024).

# Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

## 4.6 Diccionario de datos

En la base de datos MySQL, se diseñó una tabla llamada **Licencias** que contiene dos columnas principales: **serial** y **fecha**. La columna serial sirve como clave primaria, lo que garantiza que cada serial registrado en la tabla sea único. Inicialmente, cuando se añade un nuevo serial a la tabla, la columna fecha se establece por defecto en '0000-00-00', indicando que aún no se ha determinado una fecha de vencimiento para el serial. La primera vez que se ingresa un serial en la base de datos, se activaba un proceso que actualiza automáticamente la columna fecha con la fecha de vencimiento correspondiente, basándose en criterios preestablecidos.

**Figura 10.** Diccionario de datos

### datos

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	Tipo de medio
Serial	varchar(20)	No				
Fecha	date	No				

Fuente: Propia del autor (2024)

## 4.7 Plan de pruebas

Se realizaron pruebas de funcionalidad al aplicativo, con el fin de revisar y ajustar el proceso de ofuscación (Tabla 3).

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

Tabla 10. Plan de pruebas

Módulo de prueba	Objetivo de la prueba	Responsable de la prueba	Resultado de la prueba	Resultados esperados
Acceso al serial	Verificar la funcionalidad del registro a la base de datos y acceso al serial para garantizar que los usuarios puedan acceder de manera rápida a la aplicación.	Duván Stiven Garzón Morales, Andrés Felipe Gutiérrez Ceballos	<b>Prueba 1:</b> Valida correctamente que todos los campos sean llenados por el usuario.	Que el usuario se registre por primera vez de una forma correcta y a la hora de presentar el serial este sea de uso exclusivo para el usuario y no presente errores.
	verificar su capacidad para ofuscar strings y crear módulos falsos de manera efectiva, garantizando la preservación de la funcionalidad del programa original. Esto implica integrar	Duván Stiven Garzón Morales, Andrés Felipe Gutiérrez Ceballos.	<b>Prueba 1:</b> verificar su capacidad para ofuscar strings y crear módulos falsos de manera efectiva, garantizando la preservación de la	<b>Resultado 1:</b> Que el proceso de ofuscación dificulte significativamente la capacidad de los adversarios para realizar ingeniería inversa en el código del programa.

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

Módulo de prueba	Objetivo de la prueba	Responsable de la prueba	Resultado de la prueba	Resultados esperados
Funcionalidad Del Backend	cambios en los archivos mencionados de forma coherente, sin afectar su operatividad, mientras se dificulta la comprensión del código para terceros.		funcionalidad del programa original.	<b>Resultado 2:</b> Tras la ofuscación de strings y la creación de módulos falsos, se espera que el programa presente una mayor resistencia a la ingeniería inversa, dificultando la extracción de información sensible.

Fuente: Propia del autor

## CAPITULO V

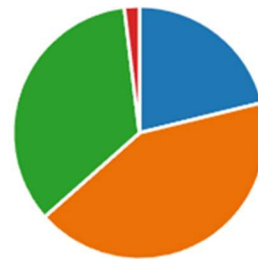
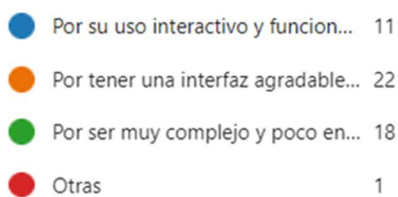
### 5.1 Análisis de datos

A continuación, se presentan los resultados de la aplicación de las encuestas con sus respectivas gráficas y análisis de resultados:

**Figura 11.** Pregunta N°1

1. ¿Cómo piensas que el software puede ayudar en la transformación digital de las empresas?

[Más detalles](#)



Fuente: Propia del autor (2024)

Analizando las respuestas de la primera pregunta (Figura 11), se pueden identificar algunos patrones y tendencias a partir de las opiniones de los encuestados con la ayuda del software para la transformación de las empresas de desarrollo de software de la ciudad de Villavicencio.

Primero, una gran mayoría de los encuestados (39/51) mencionó que la agradable interfaz de usuario del software es una forma en la que contribuye a la transformación digital de las empresas de desarrollo de software. Esto demuestra que los usuarios valoran la facilidad de uso y

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

la experiencia del usuario como características importantes del software en el contexto de la transformación digital.

En segundo lugar, un patrón identificado en las respuestas es la preocupación de los encuestados por la complejidad del software (26/51 respuestas). Los encuestados señalan que el software puede resultar difícil de entender para el usuario promedio, lo que puede ser un obstáculo para su introducción y uso efectivo en la transformación digital de las empresas de desarrollo de software. En tercer lugar, un número importante de encuestados (16/51) mencionan que el uso interactivo y la operación precisa del software son importantes para la transformación digital de las empresas de desarrollo de software. Esto demuestra que los usuarios aprecian la funcionalidad y eficiencia del software como herramienta para la transformación digital.

En conclusión, el análisis de las respuestas revela que la interfaz de usuario amigable, la complejidad del software, así como su uso interactivo y operación precisa, son aspectos cruciales para la transformación digital de las empresas de desarrollo de software en Villavicencio. Estos hallazgos resaltan la importancia de diseñar herramientas tecnológicas que sean intuitivas, funcionales y eficientes para garantizar el éxito en la adopción y el aprovechamiento de la transformación digital tanto para desarrolladores independientes como para las empresas de desarrollo de software de la ciudad de Villavicencio.

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

**Figura 12.** Pregunta N°2

2. Usted como usuario ¿Qué canales de venta considera al momento de comprar productos informáticos (Antivirus, Programas Ofimáticos, Herramientas específicas)?

[Más detalles](#)




Fuente: Propia del autor (2024)

El análisis de las respuestas revela una clara preferencia por parte de los usuarios hacia la compra de productos informáticos a través de páginas web o tiendas virtuales, siendo esta la opción más popular con el 76% de las respuestas (Figura 12). Este dato sugiere una tendencia hacia la comodidad y la conveniencia que ofrecen las plataformas en línea para adquirir este tipo de productos. Además, se observa un interés minoritario pero significativo en la compra a través de agentes externos y la venta directa con el desarrollador, lo que podría indicar una demanda de experiencias de compra más personalizadas o relaciones directas con los proveedores.

**Figura 13.** Pregunta N°3

3. ¿Del producto informático seleccionado anteriormente cumple con sus expectativas?

[Más detalles](#)

 Información



Fuente: Propia del autor (2024)

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

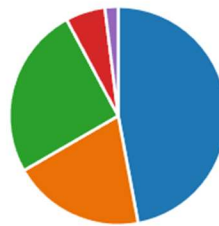
De las 50 respuestas restantes, 46 (92%) indicaron que el producto informático seleccionado cumple con las expectativas de los encuestados, mientras que 4 (8%) indicaron que no cumple con sus expectativas (Figura 13). Aunque la muestra sigue siendo relativamente pequeña, estos resultados sugieren que la mayoría de los encuestados están satisfechos con el producto informático en cuestión. Sin embargo, como se mencionó anteriormente, sería útil investigar más a fondo las razones detrás de las respuestas negativas y cómo se podrían abordar para mejorar el producto.

**Figura 14.** Pregunta N°4

4. ¿En qué área o aplicación específica planeas utilizar nuestro producto y qué características te parecen más importantes para esa aplicación en particular?

[Más detalles](#)

● Educación	24
● TI (tecnología de la información)	10
● Finanzas	13
● Fabricación	3
● Otras	1



Fuente: Propia del autor (2024)

De las 51 respuestas, la mayoría de los encuestados, que incluyen desarrolladores independientes y empresas de desarrollo de software, planean utilizar el producto principalmente en los sectores de educación y finanzas, seguidos por tecnología de la información y fabricación, aunque esta última opción fue menos mencionada (Figura 14). No se proporcionaron detalles sobre las características específicas deseadas, se puede suponer que los encuestados valoran aspectos como facilidad de uso, eficiencia, precisión y fiabilidad en sus aplicaciones.

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

**Figura 15.** Pregunta N°5

5. ¿En tu opinión, las empresas líderes en tecnología están invirtiendo suficientemente en investigación y desarrollo para continuar innovando y lanzando nuevas tecnologías al mercado?

[Más detalles](#)

● Si y mucho	35
● Medio, porque se concentran d...	14
● No para nada	1



Fuente: Propia del autor (2024)

De las 50 respuestas recibidas, el 82% considera que las empresas líderes en tecnología están invirtiendo suficientemente en investigación y desarrollo para continuar innovando y lanzando nuevas tecnologías al mercado (Figura 15). Entre quienes respaldan esta inversión, el 45% la califica como "sí y mucho", destacando la constante innovación. Sin embargo, el 14% opina que se invierte en exceso, centrando demasiado en la investigación sin aplicación práctica, mientras que el 4% considera que la inversión es insuficiente, con más enfoque en publicidad y ventas que en innovación. Los críticos de la inversión excesiva señalan la falta de aplicación práctica y experiencia del usuario, mientras que los críticos de la insuficiente inversión mencionan el enfoque excesivo en publicidad y ventas en detrimento de la investigación y el desarrollo.

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

**Figura 16.** Pregunta N°6

6. ¿Usted como usuario, Considera que este producto informatico mejore o ayude a las empresas tecnologicas?

[Más detalles](#)

[Información](#)

● Si y mucho	29
● No para nada	14
● Seguiran iguales	8



Fuente: Propia del autor (2024)

La encuesta muestra que la mayoría de los encuestados, aproximadamente el 57%, perciben que el producto informático contribuye significativamente al desarrollo de las empresas de desarrollo de software de Villavicencio, mientras que un 27% tiene una visión negativa al respecto (Figura 16). Sin embargo, un pequeño porcentaje (16%) considera que estas empresas de desarrollo de software permanecerán en un estado similar. Estos resultados sugieren una división de opiniones entre los desarrolladores y empresas de desarrollo de software, con una inclinación general hacia una percepción positiva o al menos neutral sobre el impacto del producto en el panorama tecnológico local de Villavicencio.

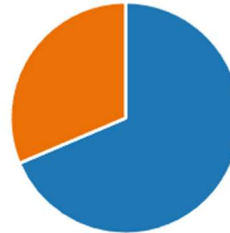
## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

**Figura 17.** Pregunta N°7

7. De este producto informatico, ¿usted considera que aporta ciertos conocimientos en el desarrollo de software?

[Más detalles](#)

🔍 Información



Fuente: Propia del autor (2024)

El análisis de datos revela que la mayoría de los desarrolladores encuestados (35 de 51) consideran que el producto informático aporta conocimientos en el desarrollo de software, mientras que 16 desarrolladores opinan lo contrario (Figura 17). Este hallazgo sugiere que la percepción de los beneficios del producto puede variar según el contexto o las necesidades específicas de la empresa de desarrollo de software. En resumen, estos datos subrayan la importancia de considerar múltiples perspectivas al evaluar la efectividad y utilidad de un producto informático.

**Figura 18.** Pregunta N°8

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

8. ¿Podría usted decir que este producto tiene rendimientos informáticos que aporten a empresas líderes en tecnología?

[Más detalles](#)



Fuente: Propia del autor (2024)

Según los datos recopilados, la mayoría de los desarrolladores, representados por un 64,7%, creen que el producto ofrece rendimientos informáticos que son beneficiosos para las empresas de desarrollo de software líderes en Villavicencio. Sin embargo, un 35,3% de los Desarrolladores expresaron la opinión contraria, argumentando que el producto carece de dichos rendimientos (Figura 18). Algunos señalan que la utilidad del producto para estas empresas de desarrollo de software depende de sus necesidades específicas y de cómo se utilicen ciertas características de este. Por otro lado, hay quienes sostienen que el producto no aporta valor adicional ni innovación significativa para este tipo de stacks.

## **CAPITULO VI**

### **6.1 Conclusiones**

Se establecieron requerimientos funcionales y no funcionales sólidos, proporcionando una base clara para el desarrollo del aplicativo de escritorio. Estos requisitos sirvieron de guía para las fases subsiguientes del proyecto. En la práctica profesional, se llevó a cabo un estudio técnico en la línea productiva enfocada en evaluar la viabilidad del Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET. Este estudio proporcionó una base para comprender los requerimientos y desafíos asociados con la creación del ofuscador, asegurando su relevancia tanto para desarrolladores independientes como para empresas de desarrollo de software en la ciudad de Villavicencio. Este análisis proporcionó una base sólida para el diseño e implementación del aplicativo, asegurando su relevancia en el contexto de los desarrolladores independientes y empresas de desarrollo de software de la ciudad de Villavicencio.

En Colombia, existe un nicho de mercado en crecimiento para herramientas de ofuscación y seguridad, tanto para empresas y organizaciones que necesitan proteger sus aplicativos. Este contexto justifica la importancia de realizar estudios técnicos y encuestas para evaluar la viabilidad y el mercado objetivo de soluciones como el ofuscador.

La encuesta virtual llevada a cabo como parte del estudio permitió recopilar opiniones, experiencias y datos estadísticos relevantes de microempresas en Villavicencio. Estos resultados no solo sirvieron como base para el desarrollo del software, sino que también proporcionan información para empresas interesadas en implementar soluciones de seguridad en la región.

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

La implementación de un algoritmo de ofuscación fue un componente clave en la protección de la propiedad intelectual y la seguridad de los archivos .NET en los aplicativos de escritorio. Este algoritmo demostró ser efectivo en la prevención de la ingeniería inversa y la protección de la propiedad intelectual de los desarrolladores y empresas de Villavicencio.

Además, La implementación de un serial único para cada usuario, junto con la ofuscación de strings y la creación de módulos falsos, fue fundamental para proteger la propiedad intelectual y prevenir posibles vulnerabilidades de seguridad en el proyecto. Estas medidas demostraron ser efectivas en la prevención de la decompilación, ingeniería inversa y crackeo del código, asegurando la integridad y confidencialidad de los datos.

### **6.2 Recomendaciones**

**Utilizar el cifrado de archivos:** Aprovecha esta función para proteger tus datos durante el desarrollo de software.

**Configurar permisos de acceso adecuados:** Establece una jerarquía de acceso cuidadosamente diseñada para restringir quién puede acceder a los archivos protegidos. Esta medida no solo se trata de controlar quién puede ver o modificar los archivos, sino también de garantizar que solo aquellos con la autorización adecuada puedan acceder a ellos.

**Realizar copias de seguridad regulares:** La realización de copias de seguridad periódicas es esencial para garantizar la disponibilidad y la integridad de los datos en caso de fallos, ya sea debido a errores humanos, problemas técnicos o ataques cibernéticos.

**Educar sobre seguridad informática:** La capacitación y la concienciación en seguridad informática son fundamentales para fortalecer la postura de seguridad de una organización.

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

Proporcionar formación regular a los usuarios sobre las últimas amenazas cibernéticas, técnicas de ingeniería social y mejores prácticas de seguridad puede ayudar a prevenir ataques exitosos como el phishing o la ingeniería social.

### 6.3 Resumen Analítico Especializado – RAE

**Tabla 11.** Formato RAE

1. Título	Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET
2. Autores	Duván Stiven Garzón Morales Andrés Flipe Gutiérrez Ceballos
3. Fecha	mayo de 2024
4. Palabras Claves	Ofuscación, Aplicación De Escritorio, Seguridad Informática
5. Descripción	Se desarrolló una aplicación de escritorio como herramienta para brindar seguridad a los archivos .exe, .dll y aplicativos desarrollados en .NET.
6. Problema	¿De qué manera se puede desarrollar una aplicación para ofuscar código fuente de proyectos realizados en .NET por desarrolladores de software de la ciudad de Villavicencio?
7. Objetivo	Desarrollar una herramienta para ofuscación de código fuente de proyectos de software dirigida a desarrolladores de Villavicencio.

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

8. Conclusiones	<p>Se establecieron requerimientos funcionales y no funcionales sólidos, proporcionando una base clara para el desarrollo del aplicativo de escritorio. Estos requisitos sirven de guía para las fases subsiguientes del proyecto.</p> <p>la implementación de un serial único para cada usuario, junto con la ofuscación de strings y la creación de módulos falsos, ha sido fundamental para proteger la propiedad intelectual y prevenir posibles vulnerabilidades de seguridad en el proyecto. Estas medidas han demostrado ser efectivas en la prevención de la decompilación, ingeniería inversa y crackeo del código, asegurando la integridad y confidencialidad de los datos.</p>
9. Autor RAE	Duván Stiven Garzón Morales Andres Flipe Gutierrez Ceballos
10. Fecha creación de RAE	20 marzo Del 2024

Fuente: Propia del autor (2024)

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

### Referencias

VMProtect Software. (2024). *VMProtect*. Obtenido de VMProtect:

<https://vmpsoft.com/vmprotect/user-manual#analysis-cracking-and-protection-of-software>

.NET Reactor Eziriz. (2023). *EZIRIZ*. Obtenido de EZIRIZ:

[https://www.eziriz.com/dotnet\\_reactor.htm](https://www.eziriz.com/dotnet_reactor.htm)

Alvarez, D. V. (22 de Diciembre de 2022). *Periodico El Tiempo*. Obtenido de Periodico

El Tiempo: <https://www.eltiempo.com/salud/hackeo-sanitas-esta-es-la-informacion-que-filtraron-los-hackers-de-la-eps-727745>

Betancourt, A. (30 de Noviembre de 2022). *Enter.co*. Obtenido de Enter.co:

<https://www.enter.co/colombia/sanitas-y-colsanitas-continuan-sin-servicios-digitales-tras-ataque-cibernetico/>

cecilphillip. (2013). *Stack Overflow*. Obtenido de Stack Overflow:

<https://stackoverflow.com/questions/14758455/faking-a-generic-method-fakeiteasy>

Code Maze . (2023). *Code Maze* . Obtenido de Code Maze : [https://code-](https://code-maze.com/csharp-string-encryption-decryption/)

[maze.com/csharp-string-encryption-decryption/](https://code-maze.com/csharp-string-encryption-decryption/)

Congreso De La Republica De Colombia. (21 de Agosto de 1999). *Sistema Unico De*

*Informacion Normativa*. Obtenido de

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=4276>

## **Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET**

Congreso De La Republica De Colombia. (2008). *Sistema Unico De Informacion*

*Normativa* . Obtenido de Ley 1266 De 2008 - Suin - Juriscol: <https://www.suin-juriscol.gov.co/viewDocument.asp?ruta=Leyes/1676616>

Fluid Attacks. (2022). *Asociación Colombiana de Informática Sistemas y Tecnologías*

*Afines*. Obtenido de Asociación Colombiana de Informática Sistemas y Tecnologías Afines: <https://acis.org.co/portal/content/la-importancia-de-evaluar-el-c%C3%B3digo-fuente-c%C3%B3mo-mantener-el-desarrollo-seguro-de-las>

Kaspersky Corporation. (2023). *Kaspersky Latam*. Obtenido de Kaspersky Latam:

[https://latam.kaspersky.com/about/press-releases/2023\\_los-productos-de-kaspersky-demuestran-100-por-ciento-de-eficacia-antimalware-en-la-prueba-de-se-labs](https://latam.kaspersky.com/about/press-releases/2023_los-productos-de-kaspersky-demuestran-100-por-ciento-de-eficacia-antimalware-en-la-prueba-de-se-labs)

Kaspersky Latam. (2021). *Kaspersky Latam*. Obtenido de

<https://latam.kaspersky.com/resource-center/definitions/encryption>

Larry, M. R. (2009). *Estadística* (4ta ed.). Mc Graw-Hill.

Maxwell, A. (Octubre de 2017). *Torrent Freak*. Obtenido de Torrent Freak:

<https://torrentfreak.com/assassins-creed-origin-drm-hammers-gamers-cpus-171030/>

Microsoft. (2023). *Encrypting data*. Obtenido de [https://learn.microsoft.com/es-](https://learn.microsoft.com/es-es/dotnet/standard/security/encrypting-data)

[es-es/dotnet/standard/security/encrypting-data](https://learn.microsoft.com/es-es/dotnet/standard/security/encrypting-data)

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

Microsoft Corporation. (2022). *Entity Framework | Microsoft Learn*. Obtenido de Entity Framework | Microsoft Learn: <https://learn.microsoft.com/en-us/aspnet/entity-framework>

Microsoft Corporation. (2022). *Learn .NET By Microsoft*. Obtenido de Learn .NET By Microsoft: <https://learn.microsoft.com/es-es/dotnet/api/system.security.cryptography.protecteddata?view=dotnet-plat-ext-7.0>

Pipelines, N. (2023). *Digital.ai*. Obtenido de Digital.ai: <https://digital.ai/es/catalyst-blog/why-code-obfuscation-matters/>

VMProtect Software. (2023). *VMProtect*. Obtenido de VMProtect: <https://vmpsoft.com/>

YAK Pro WPS. (Octubre de 2022). *Yet Another Killer Product*. Obtenido de Yet Another Killer Product: <http://webportal.yakpro.com/>

# Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

## Anexos

Anexo 1. Manual de usuario del Encriptador de archivos de desarrollo con herramientas .NET

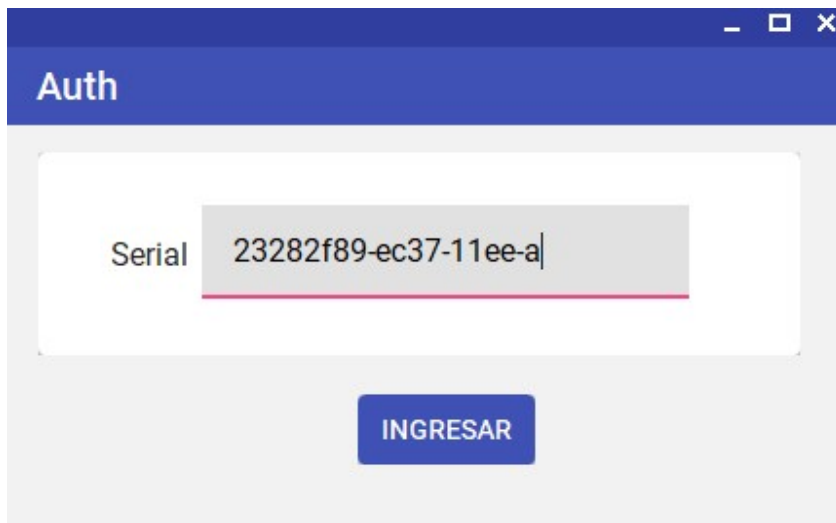
### Manual de Usuario

#### 7.1 Inicio de sesión

Para iniciar la aplicación, debe ejecutar el archivo MainChiwiro.exe, Posteriormente deberá ingresar el serial proporcionado por el soporte. En nuestro caso, será 23282f89-ec37-11ee-a.

Después de ingresar el serial, presione el botón "Ingresar", Una vez ingresado aparecerá el siguiente menú:

**Figura 19.** Formulario de registro



The image shows a screenshot of a Windows application window titled "Auth". The window has a blue header bar with the text "Auth". Below the header, there is a light gray background containing a white rounded rectangle. Inside this rectangle, the word "Serial" is followed by a text input field containing the alphanumeric string "23282f89-ec37-11ee-a". Below the input field, there is a blue button with the text "INGRESAR" in white capital letters.

Fuente: Propia del autor (2024)

## Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

### 7.2 Cargar Archivo y Ofuscar

Seleccione la opción "Cargar Archivo" para cargar el índice que desea proteger, Una vez seleccionado, presione el botón "Ofuscar" para iniciar el proceso de protección del archivo ejecutable.

**Figura 20:** Menú principal

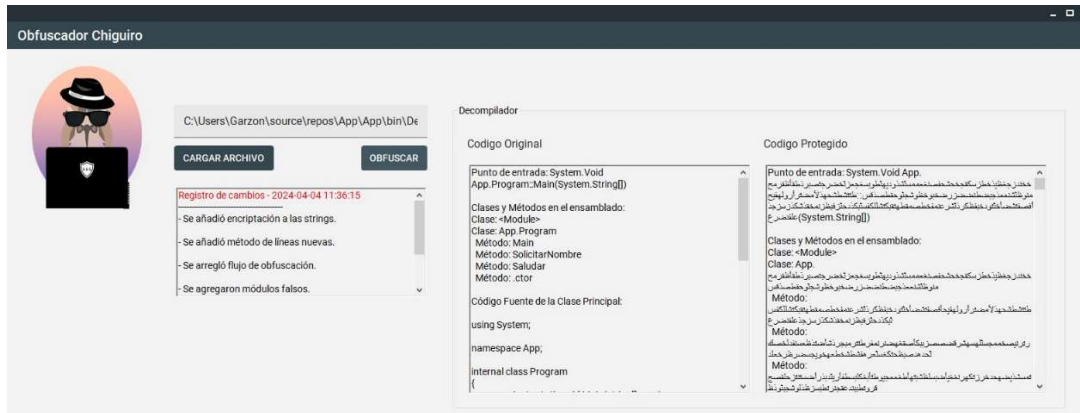


Fuente: Propia del autor (2024)

Una vez completado el proceso, se abrirá una consola con el siguiente mensaje:  
Modificación completa.

# Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

Figura 21. Ofuscación



Fuente: Propia del autor (2024)

## 7.3 Resultado final

El archivo protegido estará ubicado en la misma carpeta que el ofuscador, con el nombre Unprotected.exe. Para mayor seguridad, el archivo ha sido protegido, como se muestra en la imagen adjunta, Se han generado nuevos módulos falsos y las cadenas de texto ahora no son legibles a simple vista.

# Desarrollo de una aplicación de escritorio para la ofuscación de código fuente de proyectos realizados en .NET

Figura 22. Vista previa archivo ofuscado

```
1 namespace App
2 {
3     // Token: 0x00000002 RID: 2 Internal class Program
4     internal class Program
5     {
6         // Token: 0x00000001 RID: 1 RVA: 0x0000124C File Offset: 0x0000044C
7         private static void Main()
8         {
9             Console.WriteLine("Inicio de la aplicación");
10            for (i;
11            {
12                string = " ";
13            }
14            Console.WriteLine("Fin de la aplicación");
15        }
16    }
17 }
18
19 // Token: 0x00000002 RID: 2 RVA: 0x00001278 File Offset: 0x00000478
20 private static string Console.WriteLine()
21 {
22     Console.WriteLine(" ");
23 }
24
25 // Token: 0x00000001 RID: 1 RVA: 0x00001204 File Offset: 0x00000404
26 private static void Console.WriteLine(string)
```

Fuente: Propia del autor (2024)