

# ESCUELA POLITÉCNICA NACIONAL

## FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UNA APLICACIÓN BANCARIA QUE GENERA  
USUARIOS Y CLAVES SEGURAS SIGUIENDO LOS ESTANDARES  
DE LA EMPRESA

IMPLEMENTACIÓN Y PRUEBAS DE LA APLICACIÓN BANCARIA  
QUE GENERE USUARIOS Y CLAVES SEGURAS SIGUIENDO LOS  
ESTÁNDARES DE LA EMPRESA

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA EN  
CIENCIAS DE LA COMPUTACIÓN

ELSA MARGOTH VASCO GUALOTUÑA

[elsa.vasco@epn.edu.ec](mailto:elsa.vasco@epn.edu.ec) / [elsa.vasco01@gmail.com](mailto:elsa.vasco01@gmail.com)

DIRECTOR: Msc. María Monserrate Intriago Pazmiño

[monserrate.intriago@epn.edu.ec](mailto:monserrate.intriago@epn.edu.ec)

DMQ, agosto 2023

## **CERTIFICACIONES**

Yo, VASCO GUALOTUÑA ELSA MARGOTH declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

---

**VASCO GUALOTUÑA ELSA MARGOTH**

Certifico que el presente trabajo de integración curricular fue desarrollado por VASCO GUALOTUÑA ELSA MARGOTH, bajo mi supervisión.

---

**Msc. María Monserrate Intriago Pazmiño**  
**DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

ELSA MARGOTH VASCO GUALOTUÑA

Msc. MARÍA MONSERRATE INTRIAGO PAZMIÑO

## ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	i
<b>Error! Marcador no definido.</b>	
DECLARACIÓN DE AUTORÍA.....	<b>¡Error! Marcador no definido.</b>
ÍNDICE DE CONTENIDO.....	III
RESUMEN.....	IV
ABSTRACT.....	VI
1 INTRODUCCIÓN .....	1
1.1 Objetivo general.....	1
1.2 Objetivos específicos .....	1
1.3 Alcance .....	2
1.4 Marco teórico .....	2
2 METODOLOGÍA .....	11
2.1 Enfoque iterativo-incremental.....	11
2.2 Herramientas de Software.....	12
2.3 Gestores de Bases de Datos .....	16
2.4 Justificación de herramientas a utilizar.....	18
2.5 Implementación del proyecto .....	19
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	34
3.1 Resultados.....	34
3.2 Conclusiones .....	53
3.3 Recomendaciones .....	54
4 REFERENCIAS BIBLIOGRÁFICAS .....	55

## RESUMEN

En un mundo cada vez más digitalizado, la seguridad de datos se ha convertido en una preocupación central para individuos y organizaciones por igual. La información sensible y confidencial, como datos personales, financieros y corporativos, es un activo valioso que necesita protección robusta. En este contexto, la encriptación se ha erigido como una herramienta esencial en la salvaguardia de la integridad y privacidad de estos datos.

### Importancia de la Encriptación

La encriptación es el proceso de convertir información legible en un formato ilegible, a menos que se posea la clave de desencriptación. Esto asegura que incluso si los datos caen en manos equivocadas, resulta incomprensible y, por lo tanto, inútil. La encriptación no solo evita el acceso no autorizado, sino que también protege contra la interceptación y el robo de datos durante su transmisión. Es un pilar fundamental para mantener la confidencialidad y la integridad de la información sensible.

### Ejemplos de Incidentes de Seguridad Recientes y su Potencial Mitigación

- 1. Ataque a la cadena de suministro de software:** En el incidente de SolarWinds en 2020, se infiltraron sistemas mediante una actualización de software comprometida. Aquí, la encriptación de extremo a extremo habría limitado el acceso a datos sensibles incluso si los atacantes hubieran conseguido infiltrarse en la red.
- 2. Algoritmo Utilizado:** En este escenario, un algoritmo criptográfico de clave pública como RSA podría haber sido eficaz. Aunque es relativamente lento en comparación con otros métodos, su robustez en términos de seguridad habría ayudado a prevenir la divulgación no autorizada.
- 3. Ataques de ransomware:** Incidentes como el ataque a Colonial Pipeline en 2021 resaltan la importancia de la encriptación en reposo. Si los datos

se mantuvieran encriptados en reposo, incluso si los atacantes lograran acceder a los sistemas, no podrían acceder a la información sin la clave de desencriptación.

- 4. Algoritmo Utilizado:** AES (Advanced Encryption Standard) es un ejemplo destacado para la encriptación en reposo. Es altamente eficiente en términos de velocidad y seguridad, lo que lo hace adecuado para proteger grandes volúmenes de datos almacenados.

**PALABRAS CLAVE:**

Scrum

Usuario

Visual Studio

#C

MySql

## ABSTRACT

In an increasingly digitized world, data security has become a central concern for individuals and organizations alike. Sensitive and confidential information, such as personal, financial, and corporate data, is a valuable asset that requires robust protection. In this context, encryption has emerged as an essential tool in safeguarding the integrity and privacy of this data.

### **Importance of Encryption:**

Encryption is the process of converting readable information into an unreadable format unless one possesses the decryption key. This ensures that even if the data falls into the wrong hands, it remains incomprehensible and thus useless. Encryption not only prevents unauthorized access but also safeguards against data interception and theft during transmission. It is a fundamental pillar for maintaining the confidentiality and integrity of sensitive information.

### **Examples of Recent Security Incidents and Their Potential Mitigation:**

- 1. Software supply chain attack:** In the 2020 SolarWinds incident, systems were infiltrated through a compromised software update. Here, end-to-end encryption would have limited access to sensitive data even if attackers had managed to infiltrate the network.
- 2. Algorithm Used:** In this scenario, a public key cryptographic algorithm like RSA could have been effective. Although relatively slow compared to other methods, its security robustness would have helped prevent unauthorized disclosure.
- 3. Ransomware attacks:** Incidents like the 2021 Colonial Pipeline attack highlight the importance of encryption at rest. If data were kept encrypted at rest, even if attackers gained access to the systems, they wouldn't be able to access the information without the decryption key.
- 4. Algorithm Used:** AES (Advanced Encryption Standard) is a prominent example for encryption at rest. It is highly efficient in terms of speed and security, making it suitable for protecting large volumes of stored data.

# **1 INTRODUCCIÓN**

En un sistema bancario se requiere transportar información sensible que corresponde a las claves de usuarios internos de las aplicaciones en producción y pruebas.

Estas claves son almacenadas en una caja fuerte de manera física dentro de las instalaciones de la institución y a las cuales solo puede acceder personal autorizado.

Se requiere generar un usuario con parámetros estandarizados bajo las normas de seguridad interna de la institución y las claves de manera automática. Las mencionadas claves serán procesadas mediante algoritmos de encriptación.

Estas fases son analizadas desde el punto de vista del desarrollo del software con calidad de seguridad, específicamente la encriptación acordados en el campo científico.

En este proyecto se propone crear una aplicación de escritorio que permita la generación de usuarios con sus respectivas claves y la gestión de información necesaria para el correcto funcionamiento de esta aplicación. Se contempla estudiar las tecnologías convenientes para el desarrollo de la aplicación.

Luego de esta fase se procederá a desarrollar la aplicación mediante las herramientas de desarrollo y técnicas de encriptación.

## **1.1 Objetivo general**

Desarrollar una aplicación que permita generar usuarios, claves seguras y el guardado de la misma bajo encriptación.

## **1.2 Objetivos específicos**

- 1.- Implementar los módulos de Usuario Caja Fuerte, Usuario Genérico Sin Clave, Usuario de Aplicativo con las siguientes características: Registro, actualización y consulta.
- 2.- Realizar pruebas del guardado de la clave encriptada en la base de datos.
- 3.- Generar informe sobre datos del usuario y la clave generada.



### 1.3 Alcance

Estudio de las tecnologías adecuadas para resolver esta necesidad.

Desarrollo de la aplicación siguiendo el enfoque ágil.

Pruebas de la aplicación.

### 1.4 Marco teórico

La protección de claves utilizadas en aplicaciones bancarias es esencial para garantizar la seguridad de la información financiera a continuación algunas medidas que se pueden tener en consideración para proteger estas claves [1]:

**1.- Almacenamiento seguro de claves:** Las claves utilizadas en aplicaciones bancarias deben almacenarse de manera segura. Es recomendable utilizar técnicas de cifrado fuertes para proteger las claves almacenadas en base de datos. También se pueden emplear técnicas de hashing para almacenar las claves de forma irreversible, de manera que incluso si hay una brecha de seguridad, las claves originales no se verán comprometidas.

**2.- Acceso restringido:** Limitar el acceso a las claves solo a personal autorizado. Utilizar medidas de control de acceso, como autenticación de múltiples factores, para garantizar que solo las personas adecuadas tengan acceso a las claves. Además, es importante mantener registros de auditoría para supervisar quién accede a las claves y la fecha en que la realiza.

**3.- Gestión de claves:** Implementar una política de gestión de claves sólida. Esto incluye prácticas como la rotación regular de claves, la revocación de claves cuando sea necesario (por ejemplo: si un empleado se desvincula de la organización), y el uso de algoritmos de generación de claves seguras.

**4.- Protección de comunicación:** Asegurar que la comunicación entre la aplicación bancaria y los servidores sean seguras utilizando protocolos criptográficos como HTTPS. Esto garantizará que las claves no se

trasmítan en texto plano, lo que dificultaría su interpretación por parte de posibles ataques.

**5.- Pruebas de seguridad:** Realizar pruebas de seguridad periódicas, como pruebas de penetración y auditorías de código, para identificar posibles vulnerabilidades en la aplicación que podrían comprometer la seguridad de las claves. Corrige cualquier problema identificado de manera oportuna.

**6.- Educación y concientización:** Capacitar al personal que trabaja en claves sobre las mejores prácticas de seguridad, como la importancia de no compartir las claves, no anotarlas en lugares no seguros y estar alerta ante posibles intentos de ingeniería social o phishing.

**7.- Cumplimiento normativo:** Asegurarse de cumplir con los estándares y regulaciones de seguridad pertinentes en el ámbito bancario, con la Norma de Seguridad de Datos.

## **La criptografía**

La criptografía es una disciplina que se ocupa de asegurar la comunicación y proteger la información mediante técnicas y algoritmos matemático. Su objetivo principal es garantizar la confidencialidad, integridad, autenticidad y no repudio de los datos transmitidos o almacenados, asegurando que solo los partes autorizados puedan acceder a la información y que cualquier intento de acceso no autorizado sea detectado [2].

En resumen, la criptografía se encarga de convertir datos legibles en un formato ilegible llamado "Texto Cifrado", mediante el uso de claves o contraseñas, para que solo aquellos que posean la clave correcta puedan descifrar la información y volverla legible nuevamente.

Esta disciplina ha sido de vital importancia en la historia, especialmente en el ámbito militar y diplomático, pero con el auge de la tecnología y la digitalización, se ha vuelto esencial en el mundo de la informática y las

comunicaciones para proteger datos confidenciales, transacciones electrónicas, contraseñas, entre otros aspectos de la seguridad informática.

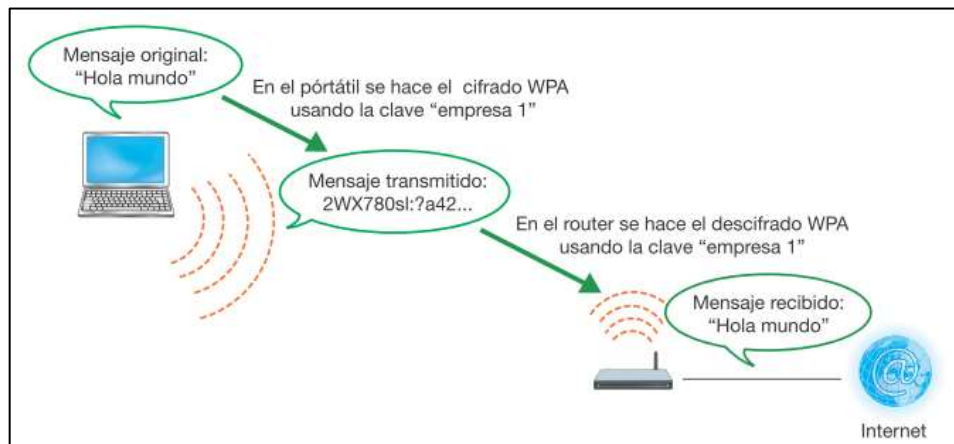


Ilustración 1: Cifrado WPA en redes wifi [3].

Las claves son una combinación de símbolos es decir letra, números, signos de puntuación, etc., por lo que nuestra seguridad está expuesta a los ataques de fuerza bruta: probar las  $n$  combinaciones posibles de símbolos. Para evitarlo se debe tomar las siguientes medidas:

- 1.- Utilizar claves de gran tamaño** (512-1024-2048-4096 bytes), de manera que el atacante necesite muchos recursos computacionales para cubrir todo el rango rápidamente.
- 2.- Modificar o cambiar la clave regularmente:** De esta forma, si alguien quiere intentar cubrir todo el rango de claves, le limitamos el tiempo para realizarlo.
- 3.- Utilizar todos los tipos de caracteres posibles:** Una clave compuesta solo de números es más fácil de descifrar que una clave compuesta de números y letras.
- 4.- No utilizar palabras fácilmente identificables:** Palabras comunes de nuestro vocabulario cotidiano, nombres propios, etc.
- 5.- Detectar repetidos intentos fallidos en un corto intervalo de tiempo:** por ejemplo, las tarjeta o móviles se bloquean después de un cierto número de intentos al introducir el PIN.

Las claves no son el único punto débil de la criptografía; pueden existir vulnerabilidades en el propio algoritmo o en la implementación del algoritmo en alguna versión de un sistema operativo o un driver concreto.

## **Criptografía simétrica y asimétrica**

### **Criptografía de clave secreta o simétrica**

La criptografía simétrica, también llamada criptografía de clave secreta, es uno de los dos principales enfoques utilizados en criptografía, siendo el otro la criptografía asimétrica (o de clave pública). En la criptografía simétrica, tanto el proceso de cifrado como el de descifrado utilizando la misma clave secreta compartida entre las partes que desean comunicarse de manera segura [3].

El funcionamiento de la criptografía simétrica es relativamente sencillo: el emisor cifra los datos utilizando la clave secreta antes de enviarlos, y el receptor utiliza la misma clave secreta para descifrar los datos y recuperar la información original. Es esencial que tanto el emisor como el receptor mantengan la clave secreta en secreto y protegida, ya que cualquier persona que tenga acceso a la clave podría descifrar los datos cifrados.

Una de las ventajas de la criptografía simétrica es que es generalmente más rápida y eficiente que la criptografía asimétrica en términos de procesamiento. Sin embargo, el mayor desafío en la criptografía simétrica es establecer y mantener la seguridad de la clave secreta compartida entre las partes. Si un atacante logra obtener la clave, podría comprometer la seguridad de la comunicación.

Por lo tanto, para garantizar la seguridad a largo plazo y proteger contra ataques potenciales, a menudo se utilizan algoritmos de cifrado simétrico en combinación de la criptografía asimétrica. En este caso, la criptografía asimétrica se utiliza para establecer una clave secreta compartida entre las partes y luego se emplea la criptografía simétrica para proteger la comunicación utilizando esa clave compartida durante la sesión.

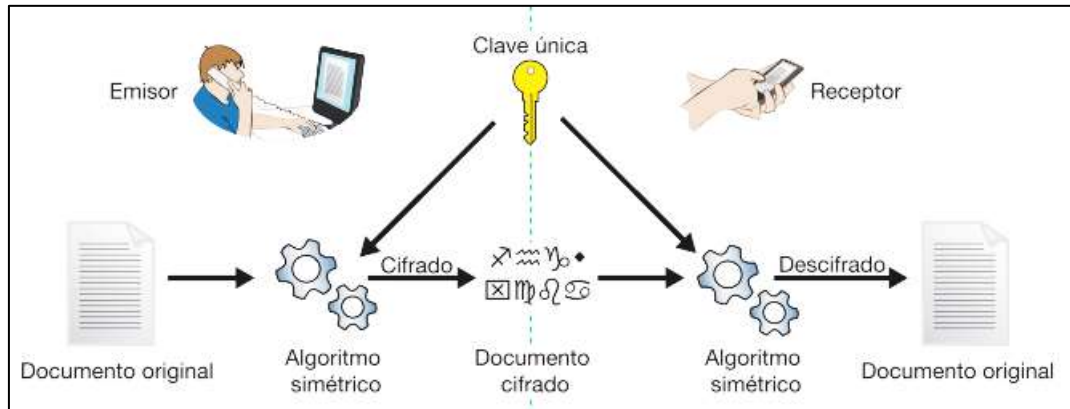


Ilustración 2: Criptografía simétrica [3].

## Criptografía asimétrica

La criptografía asimétrica, también conocida como criptografía de clave pública, es otro enfoque importante en el campo de la criptografía, complementario a la criptografía simétrica. A diferencia de la criptografía simétrica, la criptografía asimétrica utiliza un par de claves matemáticamente relacionadas pero distintas: clave pública y clave privada.

El funcionamiento de la criptografía asimétrica es el siguiente:

**1.- Clave pública:** Cada individuo tiene una clave pública que puede compartir libremente con otros. Esta clave se utiliza para cifrar los datos antes de enviarlos. Sin embargo, no se puede utilizar para descifrar la información cifrada, lo que significa que los datos cifrados solo pueden ser descifrados con la clave privada correspondiente.

**2.- Clave privada:** cada individuo también tiene una clave privada que debe mantener en secreto y proteger. Esta clave es necesario para descifrar los datos cifrados con la clave pública correspondiente. Solo el poseedor de la clave privada puede descifrar los datos y leer la información original.

Debido a la naturaleza asimétrica de las claves, la criptografía asimétrica ofrece varias ventajas importantes:

**a.- Confidencialidad:** Solo el destinatario que posee la clave privada puede descifrar los datos y leer el contenido.

**b.- Autenticación:** El emisor puede utilizar su clave privada para firmar digitalmente un mensaje, la que proporciona una prueba de que el mensaje proviene del remitente genuino.

**c.- Integridad:** Si un mensaje cifrado con la clave pública se descifra correctamente con la clave privada correspondiente, entonces se garantiza que el mensaje no ha sido alterado desde su cifrado inicial.

**d.- No repudio:** Como el proceso de firma digital vincula al remitente con el mensaje, el remitente no puede negar haber enviado el mensaje.

Aunque la criptografía asimétrica ofrece beneficios significativos en términos de seguridad y autenticación, es generalmente más lenta y requiere más recursos computacionales en comparación con la criptografía simétrica. Por lo tanto, en muchas implementaciones prácticas, se utiliza una combinación de ambos enfoques para aprovechar las fortalezas de cada uno de ellos.

### **Algoritmos de encriptación**

El algoritmo de encriptación es un procedimiento que transforma un mensaje, sin atender su estructura lingüística o significado, de tal forma que sea incomprensible, o por lo menos difícil de comprender para terceras personas.

### **Algoritmo HASH o de resumen**

Un algoritmo hash es una función matemática que toma una entrada (o mensaje) y produce una cadena de caracteres de longitud fija, que se conoce como el valor hash o resumen. El objetivo principal de un algoritmo hash es convertir datos de cualquier tamaño en una representación fija,

que generalmente es de longitud mucho más corta, pero que parece aleatoria y única para cada conjunto de datos diferente.

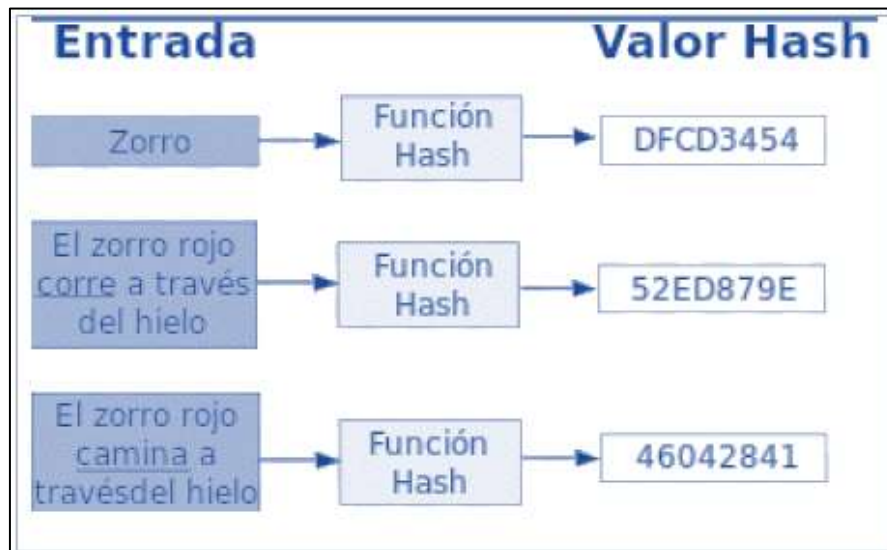


Ilustración 3: Algoritmo HASH1 [3].

Aquí hay algunas características clave de los algoritmos hash:

**Unidireccional:** Los algoritmos hash son unidireccionales, lo que significa que es fácil calcular el valor hash a partir de una entrada, pero es prácticamente imposible recuperar la entrada original a partir del valor hash.

**Determinista:** Para la misma entrada, un algoritmo hash siempre generará el mismo valor hash. Esto es importante para verificar la integridad de los datos.

**Rápido de calcular:** Los algoritmos hash están diseñados para ser eficientes en términos de cálculos, lo que los hace adecuados para una amplia gama de aplicaciones.

**Difícil de colisionar:** Una colisión ocurre cuando dos entradas diferentes generan el mismo valor hash. Los buenos algoritmos hash minimizan la posibilidad de colisiones.

**Avalancha:** Un pequeño cambio en la entrada debe generar un cambio drástico en el valor hash resultante. Esto garantiza que incluso pequeñas

diferencias en los datos de entrada produzcan valores hash completamente diferentes.

**Preimagen resistente:** Es computacionalmente difícil encontrar una entrada que genere un valor hash específico. En otras palabras, dado un valor hash, es difícil encontrar la entrada original que produjo ese valor.

**Seguro contra colisiones:** Es computacionalmente difícil encontrar dos entradas diferentes que generen el mismo valor hash.

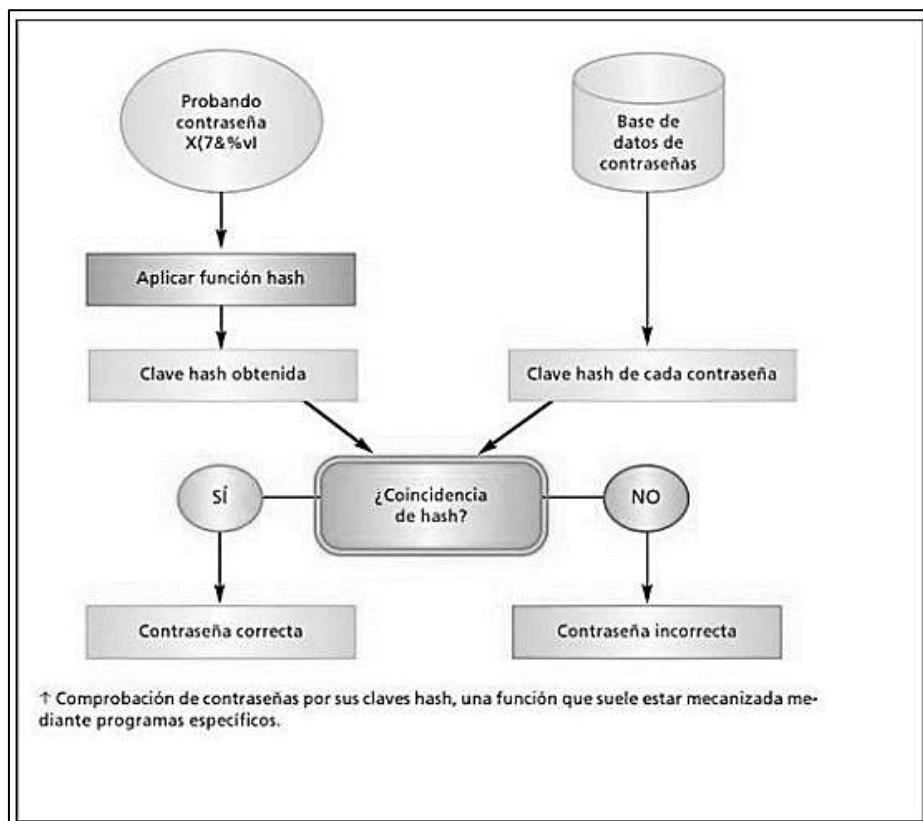


Ilustración 4: Algoritmo HASH [3].

Algunos ejemplos de algoritmos hash ampliamente utilizados son:

**MD5 (Message Digest Algorithm 5):** Aunque ampliamente utilizado en el pasado, MD5 se considera inseguro debido a vulnerabilidades que permiten colisiones.



**SHA-1 (Secure Hash Algorithm 1):** Aunque sigue siendo utilizado en algunas aplicaciones, SHA-1 también se considera inseguro debido a colisiones y esfuerzos computacionales para romperlo.

**SHA-256 (Secure Hash Algorithm 256):** Una variante más segura de la familia SHA-2. Ampliamente utilizado en aplicaciones modernas.

**SHA-3:** Una nueva familia de algoritmos hash desarrollada como una alternativa a los algoritmos anteriores. Fue diseñada para proporcionar una mayor seguridad y resistencia a ataques criptográficos.

Los algoritmos hash son esenciales en muchos aspectos de la seguridad informática, como la verificación de integridad de archivos, la protección de contraseñas y la firma digital.

## **2 METODOLOGÍA**

Para el desarrollo de este trabajo de titulación se siguió un enfoque iterativo-incremental. Con base en lo expuesto en el marco teórico se aplicará la encriptación, en donde un emisor quiere enviar información al receptor de manera segura, se lo puede realizar, sin que terceras personas sepan cual es la información que está viajando.

### **2.1 Enfoque iterativo-incremental**

El enfoque iterativo-incremental es una metodología utilizada en el desarrollo de proyectos que se caracteriza por dividir el trabajo en pequeñas etapas o iteraciones, en lugar de intentar abordar todo el proyecto de una sola vez. Cada iteración implica revisar y mejorar lo que ya se ha hecho, agregando pequeños incrementos o mejoras en cada ciclo.

Este enfoque tiene varias ventajas, como la capacidad de adaptarse a cambios en los requisitos del proyecto, la posibilidad de obtener retroalimentación temprana de los usuarios y la reducción de riesgos al abordar problemas más pequeños y manejables en lugar de enfrentar todo el proyecto de una vez.

En resumen, el enfoque iterativo-incremental es una forma efectiva de gestionar proyectos al dividirlos en partes más pequeñas y manejables, lo que facilita la adaptación a cambios y la mejora continua.

El enfoque iterativo-incremental en el desarrollo de software es un proceso de desarrollo en el que el producto se construye en una serie de iteraciones cortas, cada una de las cuales agrega una nueva funcionalidad o mejora la existente. En cada iteración, el equipo de desarrollo recopila comentarios de los usuarios para refinar el producto.

Este enfoque tiene varias ventajas, entre ellas:

- Reduce el riesgo de fracaso del proyecto: Al dividir el proyecto en iteraciones más pequeñas, el equipo puede identificar y corregir los problemas temprano en el proceso.

- Mejora la satisfacción del cliente: Al entregar versiones funcionales del producto de forma regular, los usuarios pueden ver cómo está avanzando el proyecto y proporcionar comentarios.
- Se adapta a los cambios de requisitos: Los requisitos de los usuarios pueden cambiar a medida que se desarrolla el proyecto. El enfoque iterativo-incremental permite al equipo adaptarse a estos cambios de forma rápida y sencilla.

El enfoque iterativo-incremental se utiliza en una variedad de proyectos de software, desde pequeños proyectos web hasta grandes aplicaciones empresariales. Algunas de las metodologías ágiles más populares, como Scrum y Kanban, se basan en este enfoque.

Debido a que en el desarrollo de este trabajo de titulación el equipo de trabajo está compuesto por una sola persona no se ha planteado el uso de SCRUM, XP o Kanban. Se ha optado por adoptar las iteraciones y se realizan revisiones de cada iteración con el docente director.

Algunas de las desventajas más comunes incluyen:

1. Mayor complejidad de gestión: Gestionar múltiples iteraciones y seguir el progreso en cada una puede volverse complicado, especialmente en proyectos grandes.
2. Aumento de los costos a largo plazo: Puede haber costos adicionales asociados con la gestión de Múltiples iteraciones, especialmente si se requiere una infraestructura de desarrollo y pruebas más robusta.
3. Requerimientos no definidos desde el principio: Algunos proyectos pueden beneficiar de una planificación y definición exhaustivas desde el principio, y el enfoque iterativo-incremental puede parecer ineficiente en tales casos.
4. Posibilidad de desviarse del objetivo original: Si no se gestiona adecuadamente,

## **2.2 Herramientas de Software**

Se llaman herramientas de software a los lenguajes de programación que se vaya a utilizar en el desarrollo, se trata de utilizar los más apropiados y

los que nos dé como resultado la encriptación de datos en un aplicativo de escritorio, adicional se usara los gestores de bases de datos que es fundamental del aplicativo [4].

## **Visual Studio 2022**

Visual Studio es un entorno de desarrollo integrado (IDE) creado por Microsoft que se utiliza para desarrollar aplicaciones de software, sitios web, aplicaciones móviles y otros proyectos tecnológicos. Ofrece un conjunto de herramientas y características para escribir, depurar y desplegar código de manera eficiente. Visual Studio es ampliamente utilizado por desarrolladores en todo el mundo y es compatible con varios lenguajes de programación y tecnologías [5], [6].

Algunas características y capacidades que podrían estar presentes en Visual Studio 2022 o futuras versiones podrían incluir:

**Mejoras en la productividad:** Microsoft suele trabajar en mejorar la productividad del desarrollador, mediante características como sugerencias de código, navegación rápida, refactorización de código y más.

**Compatibilidad con las últimas tecnologías:** Cada nueva versión de Visual Studio tiende a incorporar soporte para las tecnologías y plataformas más recientes, como nuevas versiones de .NET Framework, ASP.NET, C#, Python, etc.

**Integración de servicios en la nube:** Microsoft tiende a integrar sus servicios en la nube, como Azure, directamente en Visual Studio para simplificar la creación y el despliegue de aplicaciones en la nube.

**Herramientas para la colaboración en equipo:** Mejorar la colaboración en equipo es un enfoque constante, por lo que es probable que haya más características para facilitar la colaboración entre desarrolladores.

**Mejoras en la depuración y el rendimiento:** Las capacidades de depuración y perfilado suelen mejorarse en cada versión para facilitar la identificación y solución de problemas en el código.

**Interfaz de usuario y experiencia de usuario mejoradas:** Microsoft podría introducir mejoras en la interfaz de usuario y la experiencia general del usuario para hacer que la herramienta sea más intuitiva y eficiente.

**Desarrollo multiplataforma:** Dado el crecimiento del desarrollo multiplataforma, es probable que Visual Studio siga proporcionando herramientas para desarrollar aplicaciones en una variedad de plataformas, incluyendo Windows, macOS, Android, iOS, entre otros.

- **C#**

C# (pronunciado "C sharp") es un lenguaje de programación desarrollado por Microsoft como parte de su plataforma .NET. Fue diseñado para ser moderno, orientado a objetos y seguro, y es ampliamente utilizado para desarrollar una variedad de aplicaciones, desde aplicaciones de escritorio hasta aplicaciones web y móviles.

A continuación, una descripción general de algunas características y conceptos clave de C#:

**Orientado a Objetos:** C# es un lenguaje de programación orientado a objetos, lo que significa que se basa en la idea de organizar la funcionalidad en objetos que pueden contener datos y métodos. Esto permite la reutilización de código y la creación de sistemas más modulares.

**Tipo Estático:** C# es un lenguaje de tipo estático, lo que significa que los tipos de datos de las variables se comprueban en tiempo de compilación. Esto ayuda a prevenir errores de tipo comunes y mejora la seguridad.

**Plataforma .NET:** C# está estrechamente asociado con la plataforma .NET de Microsoft, que proporciona un conjunto de

bibliotecas y herramientas para el desarrollo de aplicaciones. Estas bibliotecas abarcan una amplia gama de funcionalidades, desde manipulación de datos hasta interacción con redes y sistemas.

**Sintaxis Similar a C++ y Java:** Si tienes experiencia en C++, Java u otros lenguajes C-like, encontrarás que la sintaxis de C# es bastante similar y fácil de aprender.

**Gestión de Memoria:** C# utiliza el Garbage Collector para gestionar automáticamente la memoria asignada para los objetos, lo que ayuda a evitar problemas de fugas de memoria y simplifica el desarrollo.

**Delegados y Eventos:** C# introduce conceptos como delegados y eventos, que permiten la programación orientada a eventos y son esenciales para el desarrollo de aplicaciones gráficas y basadas en interacción.

**LINQ (Language Integrated Query):** C# incluye LINQ, que permite realizar consultas a colecciones de datos directamente en el código, lo que facilita el manejo y la manipulación de datos.

**Asincronía:** C# ofrece soporte nativo para programación asíncrona y paralela, lo que permite que las aplicaciones sean más receptivas y puedan manejar tareas en segundo plano sin bloquear la ejecución principal.

**Seguridad:** C# tiene características de seguridad incorporadas, como la administración de permisos y el manejo seguro de excepciones.

**Multiplataforma:** Con la introducción de .NET Core (ahora conocido como .NET 5+), C# se ha vuelto más multiplataforma, permitiendo

desarrollar aplicaciones en sistemas operativos como Windows, macOS y varias distribuciones de Linux.

C# es utilizado en una variedad de aplicaciones, incluyendo aplicaciones de escritorio a través de Windows Presentation Foundation (WPF), aplicaciones web con ASP.NET y aplicaciones móviles con Xamarin. Es un lenguaje versátil y poderoso que sigue evolucionando con nuevas características y mejoras en cada versión.

## 2.3 Gestores de Bases de Datos

Los gestores de bases de datos son sistemas de software diseñados para administrar, almacenar y recuperar datos de manera eficiente y segura. Permiten a las organizaciones organizar grandes cantidades de información y acceder a ella de manera rápida y confiable. Hay varios tipos de gestores de bases de datos, cada uno con sus propias características y casos de uso específicos.

**Microsoft SQL Server:** Es un sistema de gestión de bases de datos desarrollado por Microsoft. Ofrece una amplia gama de características, desde gestión de datos hasta análisis de negocios y Business Intelligence. Viene en diferentes ediciones para satisfacer las necesidades de diferentes tipos de empresas.

### **MySQL**

Es un sistema de gestión de bases de datos (SGBD) de código abierto ampliamente utilizado. Es conocido por su velocidad, confiabilidad y facilidad de uso. MySQL es comúnmente utilizado en aplicaciones web y es parte fundamental de la pila LAMP (Linux, Apache, MySQL, PHP/Python/Perl).

Encriptación de datos en MySQL en ocasiones no es suficiente encriptar las comunicaciones. Si nuestros datos son altamente sensibles, se debe encriptar el mismísimo almacenamiento físico.

**Contraseña:** las contraseñas de los usuarios son datos sensibles y críticos que se deben almacenar en “claro”.

Además, tampoco sería adecuado que el administrador de la base de datos tenga la opción de conocer las contraseñas, para ello se debe guardar las contraseñas encriptadas, pero en lugar de utilizar algún algoritmo de encriptación que con el tiempo y recursos se podrá desencriptar, las contraseñas deben o preferiblemente ser almacenadas en un hash.

Al realizar el almacenamiento en hash de la contraseña, la única forma de romperlas será realizando todas las pruebas posibles hasta dar con una que genere el mismo hash. Para lo cual se requiere de tiempo y recursos.

**MySQL** tiene varias funciones de hash: PASSWORD, OLD \_PASSWORD, ENCRYPT, SHA1 y MD%

**Para la inserción de una contraseña seria:** INSERT INTO USUARIO (nombre, clave) VALUES ('NOMBRE', PASSWORD('escobar')); de esta forma, la tabla almacena el hash, no la clave.

Para el caso de aplicaciones informáticas, es preferible que el hash se realice en el cliente, evitando así que la clave en claro viaje por la red.

Comprobación de la clave de un usuario seria: SELECT \* FROM usuario WHERE nombre = \$nombre AND clave= PASSWORD(\$clave)

MySQL incluye soporte para conexiones seguras (cifrados) entre los clientes MySQL y el servidor, utilizando el protocolo SSL (Secure Sockets Layer).

La configuración de MySQL tiene la misión de ser tan rápida como le sea posible, así que no se usan las conexiones cifradas por defecto, en caso de hacerlo, haría que el protocolo cliente/servidor fuese mucho más lento.



MySQL permite que el cifrado sea activado para conexiones individuales. Puede escoger entre una conexión normal sin cifra, o una segura cifrada mediante SSL dependiendo de los requerimientos de las aplicaciones individuales.

## **2.4 Justificación de herramientas a utilizar**

La construcción del aplicativo se llevará a cabo en Visual Studio 2022 utilizando C#. Optaremos por utilizar páginas maestras (master pages) para establecer la estructura interna. Esto nos proporcionará una mayor eficiencia, ya que se caracteriza por su capacidad de respuesta ante los cambios en los requisitos que surgen debido a nuestras necesidades o a las evoluciones que experimenta nuestro proyecto.

Adicionalmente, optaremos por emplear MySQL como nuestro sistema gestor de bases de datos. Estas dos herramientas adquieren gran relevancia ya que el aplicativo que estamos por desarrollar será dinámico, lo que implica que la base de datos se actualizará de forma constante. Esto permitirá que los usuarios disfruten de información siempre actualizada tanto para los mensajes como para las claves.

En resumen, dispondremos de herramientas que se ajustarán a nuestras expectativas, otorgándonos flexibilidad para hacer frente a cambios internos y externos. Esta elección nos proporcionará una reducción en el tiempo necesario, ya que nos permitirá utilizar las funcionalidades más importantes del proyecto antes de su total finalización. Además, lograremos una mayor calidad en el software, incrementando la productividad y maximizando el retorno de la inversión.

La habilidad de realizar predicciones de tiempo nos ofrecerá una comprensión de la velocidad promedio con la que avanzamos en el desarrollo de la disertación. Para tal fin, haremos uso de puntos de historia, lo que a su vez facilitará la estimación de cuándo estará disponible una determinada funcionalidad. Al mismo tiempo, esta elección contribuirá a

reducir los riesgos, ya que nos permitirá anticipar y gestionar eficazmente cualquier desafío potencial.



*Ilustración 5: Herramientas*

## 2.5 Implementación del proyecto

Posterior a llevar a cabo el análisis y planificación de la codificación requerida y reunir la información necesaria, procedemos a ejecutar las iteraciones.

*Tabla 1: Listado de tareas para el desarrollo. Elaborado por Elsa Vasco*

Iteración	Tiempo (semanas)	Tareas
1	1	Preparación del ambiente de desarrollo front&back end
2	2	Cuenta de Usuario tipo 1 – Caja Fuerte, operaciones CRUD.
3	1	Panel Principal de Caja Fuerte
4	2	Usuario tipo 2 – Usuario Caja Fuerte
5	2	Usuario tipo 3 – Usuario Genérico sin Clave
6	2	Usuario tipo 4 – Usuario de Aplicativo
7	3	Reportes
Total	13	

## Preparación del ambiente de desarrollo front&back end

Para poder realizar el intercambio de funcionalidad entre la interfaz y la base de datos se utilizar la siguiente función *establecerConexion*

Tabla 2: Código para la conexión con la base de datos. Elaborado por Elsa Vasco

```
public void establecerConexion()
{
    MySqlConnection conex = new
    MySqlConnection(cadenaConecxion);
    try
    {
        conex.Open();
    }
    catch (Exception e)
    {
        MessageBox.Show("no se pudo conectar a la base de datos,
        error:" + e.ToString());
        return;
    }
    conex.Close();
}
```

Adicional se usará la función MySqlConnection para completar con la conexión a la Base de Datos.

Tabla 3: Datos de acceso a la Base de datos que serán llamados desde la función MySqlConnection. Elaborado por Elsa Vasco.

```
public static MySqlConnection getConexion()
{
    string servidor = "localhost";
    string bd = "cajafuerte";
    string usuario = "root";
    string password = "";
    string puerto = "3306";
    string cadenaConecxion = "server=" + servidor + ";" +
    "port=" + puerto + ";" + "user id =" + usuario + ";" + "password=" +
    password + ";" + "database=" + bd + ";";
    MySqlConnection conexion = new
    MySqlConnection(cadenaConecxion);

    return conexion;
}
```

Con las funciones mencionas anteriormente se facilita el manejo entre el aplicativo y la Base de Datos.

## Cuenta de Usuario tipo 1 – Caja Fuerte

El servicio de Caja Fuerte solamente recopila la información necesaria para llevar a cabo la autenticación, el usuario de tipo 1, es aquel que está autorizado para ingresar al sistema, y poder registrar, modificar y consultar los usuarios tipo 2, usuario tipo 3 y usuario tipo 4.

Al ingresar al sistema visualizara nuestra página de Login, que accede con las credenciales de usuario tipo 1.



*Ilustración 6: Pagina de Login del Aplicativo Caja Fuerte*

A continuación, se presenta el código en lenguaje C# que se utilizó para visualizar la ilustración 8 y su flujo para ingresar en el aplicativo.

*Tabla 4: Código de diseño y la parte lógica de la página de login. Elaborado por Elsa Vasco*

```
using System;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CajaFuerte
{
    public partial class Login : Form
```

```

{
    public Login()
    {
        InitializeComponent();
    }

    private void Login_Load(object sender, EventArgs e)
    {
        Login f1 = new Login();
        f1.CenterToScreen();
    }

    static string servidor = "localhost";
    static string bd = "cajafuerte";
    static string usuario = "root";
    static string password = "";
    static string puerto = "3306";

    string cadenaConeccion = "server=" + servidor + ";" + "port=" +
    puerto + ";" + "user id =" + usuario + ";" + "password=" + password +
    ";" + "database=" + bd + ";";

    private void btIngresar_Click(object sender, EventArgs e)
    {
        string username, pass;
        username = tbUsuario.Text;
        pass = tbContrasena.Text;
        MySqlConnection conex = new
        MySqlConnection(cadenaConeccion);

        try
        {
            conex.Open();
        }
        catch (MySqlException ex) {
            MessageBox.Show("Error " + ex.ToString());
        }

        String sql = "select usuario,contrasena from usuario
where usuario =" + username + "' AND contrasena =" + pass + "' ";
        MySqlCommand comando = new MySqlCommand(sql, conex);
        MySqlDataReader read = comando.ExecuteReader();
        if (read.Read())
        {
            this.Hide();
            Inicio inicio = new Inicio();
            inicio.Show();
        }
        else
        {
            MessageBox.Show("No existe el usuario " + username);
        }
    }

    private void btRegistrar_Click(object sender, EventArgs e)
    {
        this.Hide();
        frmRegistro registrarUsuario = new frmRegistro();
        registrarUsuario.Show();
    }
}
}

```

## Como obtener una cuenta de Usuario tipo 1

Puede crear una cuenta de inicio de sesión de la siguiente forma:

- Registrándose en el apartado Registrar que se encuentra en Login.

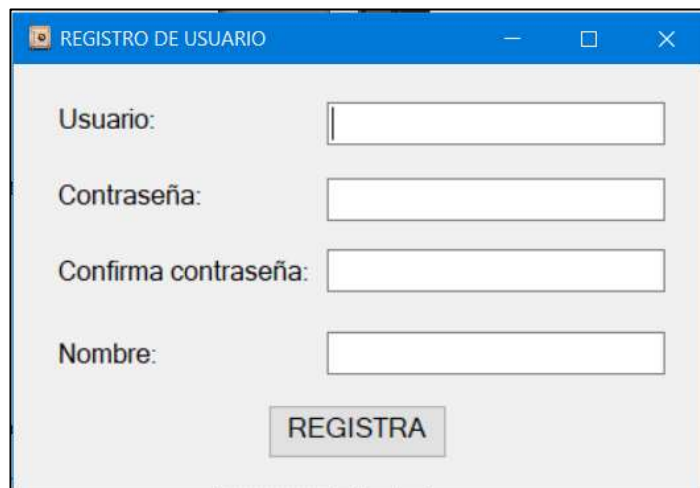


Ilustración 7: Pagina de Registro de Usuario tipo 1

Para obtener el resultado de la Ilustración 9 se utilizó el siguiente código realizado en lenguaje C#.

Tabla 5: Código en C# para el registro de usuario tipo 1. Elaborado por Elsa Vasco

```
using System;
using System.Windows.Forms;
using CajaFuerte.Clases;

namespace CajaFuerte
{
    public partial class frmRegistro : Form
    {
        public frmRegistro()
        {
            InitializeComponent();
        }

        private void btnRegistrar_Click(object sender, EventArgs e)
        {
            Usuarios usuario = new Usuarios();
            usuario.Usuario = txtUsuario.Text;
            usuario.Password = txtPassword.Text;
            usuario.ConPassword = txtConPassword.Text;
            usuario.Nombre = txtNombre.Text;
        }
    }
}
```

```

try
{
    ControlLogin control = new ControlLogin();
    string respuesta = control.ctrlRegistro(usuario);

    if (respuesta.Length > 0)
    {
        MessageBox.Show(respuesta, "Aviso",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        MessageBox.Show("Usuario registrado", "Aviso",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        this.Hide();
        Login login = new Login();
        login.Show();
    }
} catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
}
}

```

## Panel Principal de Caja Fuerte

Al ingresar al aplicativo podremos visualizar un panel con las siguientes alternativas:

- Usuario Caja Fuerte
- Usuario Genérico sin clave
- Usuario de Aplicativo
- Reportes



Ilustración 8: Panel principal Caja Fuerte

La ilustración 10 se obtiene con el siguiente código de C# en el que se desarrolla cada una de las opciones presentadas, con funciones que permiten acceder para interactuar y poder realizar el ingreso de distintos usuarios.

Tabla 6: Código en C# para la generación de la ventana principal. Elaborado por Elsa Vasco

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using CajaFuerte.Reportes;

namespace CajaFuerte
{
    public partial class Inicio : Form
    {
        public Inicio()
        {
            InitializeComponent();
        }

        private void sALIRToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            Application.Exit();
        }

        private void registroUCF_Click(object sender, EventArgs e)
        {
```



```

        UsuarioCajaFuerte lh = new UsuarioCajaFuerte();
        AddFormulario(lh);
    }

    private void registrarUGSC_Click(object sender, EventArgs e)
    {
        UsuarioGenericoSinClave lh = new
UsuarioGenericoSinClave();
        AddFormulario(lh);
    }

    private void registrarUDS_Click(object sender, EventArgs e)
    {
        UsuarioDeAplicativo lh = new UsuarioDeAplicativo();
        AddFormulario(lh);
    }

    public void AddFormulario(Form f)
    {
        if (this.panelPrincipal.Controls.Count > 0)
        {
            this.panelPrincipal.Controls.RemoveAt(0);
        }

        f.TopLevel = false;
        f.FormBorderStyle = FormBorderStyle.None;
        f.Dock = DockStyle.Fill;
        panelPrincipal.Tag = f;
        f.BringToFront();
        this.panelPrincipal.Controls.Add(f);
        f.Show();
    }

    private void bUSCARToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        DatosUCF lh = new DatosUCF();
        AddFormulario(lh);
    }

    private void bUSCARToolStripMenuItem1_Click(object sender,
EventArgs e)
    {
        DatosGSC lh = new DatosGSC();
        AddFormulario(lh);
    }

    private void bUSCARToolStripMenuItem2_Click(object sender,
EventArgs e)
    {
        DatosUS lh = new DatosUS();
        AddFormulario(lh);
    }

    private void Inicio_Load(object sender, EventArgs e)
    {
        this.FormClosed += new FormClosedEventHandler(cerrarForm);
    }

    private void cerrarForm(object sender, EventArgs e)

```

```

        {
            Application.Exit();
        }

        private void sALIRToolStripMenuItem_Click_1(object sender,
EventArgs e)
        {
            Application.Exit();
        }

        private void actaDeEntregaToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            ReporteActaRecepcionUCF lh= new ReporteActaRecepcionUCF();
            AddFormulario(lh);
        }

        private void entregaClaveCFToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            ReporteClaveUCF lh= new ReporteClaveUCF();
            AddFormulario(lh);
        }
    }
}

```

## Usuario tipo 2 – Usuario Caja Fuerte

La aplicación le permite registrar, actualizar y buscar a usuarios tipo 2, que son los usuarios principales de la aplicación de Caja Fuerte, en la que se debe registrar los datos que permitan al solicitante poder dar de baja al usuario creado o realizar alguna modificación que se necesite.

En el ámbito bancario es necesario mantener el registro de que usuarios acceden a cada aplicativo y por ello se crean usuarios con claves de acceso restringido el cual podrán ser monitoreados y se llevara el registro de este.

La clave que se entrega para el usuario tipo 2 se almacenará encriptada en BBDD y se podrá acceder a la misma al momento de actualizar los datos.

Ilustración 9: Registro de Usuario tipo 2 (Vasco Elsa, 2023)

La particularidad del form es que adjuntamos el Generador de Contraseña que permite hacer una combinación de 12 caracteres entre ellas tenemos caracteres alfabéticos (Mayúscula y minúscula), numéricas y caracteres especiales, que permitirá ingresar al usuario tipo 2 en el aplicativo solicitado.

A continuación, exponemos la función utilizada para realizar la combinación de contraseña.

Tabla 7: Código para la generación de la contraseña de forma aleatoria. Elaborado por Elsa Vasco

```
private void button1_Click(object sender, EventArgs e)
{
    var characters =
    "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz23456789@!*&%$#~=";
    var Charsarr = new char[12];
    var random = new Random();

    for (int i = 0; i < Charsarr.Length; i++)
    {
        Charsarr[i] =
        characters[random.Next(characters.Length)];
    }

    var resultString = new String(Charsarr);
    tbContraseña.Text = resultString;
}
}
```

Para realizar el almacenamiento de la contraseña generada en la BBDD se utiliza el algoritmo AES que se expone a continuación.

Tabla 8: Código de encriptación para la contraseña generada y poder almacenar de manera segura en BBDD. Elaborado por Elsa Vasco

```
public static string EncryptString(string plainText, string key)
{
    byte[] iv = new byte[16];
    byte[] array;

    using (Aes aes = Aes.Create())
    {
        aes.Key = Encoding.UTF8.GetBytes(key + "s2022");
        aes.IV = iv;

        ICryptoTransform encryptor =
        aes.CreateEncryptor(aes.Key, aes.IV);

        using (MemoryStream memoryStream = new MemoryStream())
        {
            using (CryptoStream cryptoStream = new
            CryptoStream((Stream)memoryStream, encryptor, CryptoStreamMode.Write))
            {
                using (StreamWriter streamWriter = new
                StreamWriter((Stream)cryptoStream))
                {
                    streamWriter.Write(plainText);
                }

                array = memoryStream.ToArray();
            }
        }
    }

    return Convert.ToBase64String(array);
}
```

Si nos dirigimos al apartado de BUSCAR usuario caja fuerte podremos visualizar el top 100 de los usuarios, y podremos realizar una búsqueda mas especifica por alguna característica del usuario para que sea más efectiva.

**LISTA DE USUARIOS DE CAJA FUERTE**

Parametro de Busqueda

idUsuarioCaj	codigo	usuario	clasificacion	tipoUsuario	ambiente	empresa	aplicacion	plataforma	nombreServ
31	BGR0000013	usrpsad	APLICACIÓN	LOCAL	DESARRO...	PROVEHE...	edad	LINUX	edad
32	BGR0000014	usrpsad	APLICACIÓN	LOCAL	DESARRO...	PROVEHE...	edad	LINUX	edad
33	BGR0000015	usrpsad	APLICACIÓN	DOMINIO	DESARRO...	PROVEHE...	edad	LINUX	edad
34	BGR0000016	usrpsad	APLICACIÓN	DOMINIO	DESARRO...	PROVEHE...	edad	LINUX	edad
35	BGR0000011	usrpsad	APLICACIÓN	DOMINIO	PRODUCC...	TCS	edad	ORACLE	edad
36	BGR0000012	usrpsad	APLICACIÓN	LOCAL	PRODUCC...	PROVEHE...	edad	SFTP	edad
37	BGR0000007	usrpsad	APLICACIÓN	LOCAL	TEST	PROVEHE...	edad	OTROS	edad

Ilustración 10: Lista de usuarios tipo 2 - Usuario Caja Fuerte

En el apartado de la actualización de datos de un usuario por regla del negocio bancario solo se permite modificar tres parámetros:

- Responsable Autorizador
- Observaciones
- Estado

No se permite realizar ninguna otra modificación por motivos de auditoría.

**ACTUALIZAR USUARIO CAJA FUERTE**

CÓDIGO	BGR000015	NOMBRE DEL PROYECTO	PRYxxxx
USUARIO	usrpsad	SOLICITANTE	xxxxxx
CLASIFICACIÓN DE USUARIO	APLICACIÓN	CREADO POR	CAMPANA NELLY
TIPO DE USUARIO	DOMINIO	ODT	12313
AMBIENTE	DESARROLLO	FECHA DE CREACIÓN	17/1/2022
EMPRESA	PROVEEDOR	ESTADO	ACTIVO
APLICACIÓN	edad	VIGENCIA	PERMANENTE
PLATAFORMA	LINUX	ENTREGA SOBRE RESPONSABILIDAD	SI
NOMBRE DEL SERVIDOR	edad	REGISTRO CENTRALIZADA	SI
IP DEL SERVIDOR	12313	APLICACIÓN PRODUCCIÓN	SI
RESPONSABLE XXX	12313	CLAVE	SI
CUSTODIO XXX	ADMINISTRACIÓN ENTIDAD	ACTA ENTREGA/RECEPCIÓN	SI
OBSERVACIÓN	edad	JUSTIFICACIÓN TÉCNICA	edad
GENERAR CONTRASEÑA	123456789		

Ilustración 11: Actualizar Datos Usuario Tipo 2 - Usuario Caja Fuerte

### Usuario tipo 3 – Usuario Genérico sin Clave

El usuario tipo 3 o Usuario Genérico sin Clave, este tipo de usuario es utilizado de manera temporal para acceder al aplicativo.

En el ámbito bancario es necesario tener un monitoreo constante del acceso a los aplicativos, en ocasiones por motivo de auditoria es necesario acceder a los mismo por determinados tiempos sin tener que permanecer activo el usuario con el que se accede.

Ilustración 12: Usuario tipo 3 - Usuario sin clave

Para el usuario tipo 3 disponemos de la opción de BUSCAR el cual le permite visualizar un listado al momento de ingresar y además podemos realizar búsqueda por algún parámetro determinado.

idUsuario	codigo	usuario	tipo	tipoUsuario	ambiente	empresa	aplicacion	plataforma	nombreServidor	estado
1	12	asa	adl	adad	adad	adad	ada	adad	adad	a
2	BGRSC00...									
3	BGRSC00...	ad	APLICACIÓN	LOCAL	PRODUCC...	PROVEHE...	dad	LINUX	adad	a
4	BGRSC00...									
5	BGRSC00...	afa	APLICACIÓN	DOMINIO	PRODUCC...	TCS	afaf	ORACLE	afaf	1
6	BGRSC00...	asda	BASES DE	DOMINIO	PRODUCC...	TCS	adafaf	BFM	afaf	2
7	BGRSC00...	ada	BASES DE	LOCAL	PRODUCC...	TCS	adad	BFM	adad	d
8	BGRSC00...	asafada	BASES DE	DOMINIO	PRODUCC...	PROVEHE...	asda	BFMs	asda	a
9	BGRSC00...	asafada	BASES DE	LOCAL	DESARRO...	PROVEHE...	asda	BFM	asafada	1
10	BGRSC00...	asafada	BASES DE	DOMINIO	DESARRO...	PROVEHE...	asafad	SIG	asafad	1
11	BGRSC00...	asafada	BASES DE	DOMINIO	DESARRO...	PROVEHE...	adad	BFM	adad	1
12	BGRSC00...	asafada	OTROS	DOMINIO	PRODUCC...	TCS	Buzaj	BFM	asafad	1

Ilustración 13: Lista de Usuario Genérico sin Clave

El usuario tipo 3 también dispone de la opción de actualizar Datos en el que por regla de negocio solo se puede modificar el Estado, Responsable

XXX y la Justificación, los demás datos son sensible por auditoria y no son modificables.

Ilustración 14: Actualizar Usuario tipo 3 - Usuario sin clave

### Usuario tipo 4 – Usuario de Aplicativo

El usuario tipo 4 o usuario de aplicativo es destinado para mantener el registro de y seguimientos de usuario con privilegios solo de lectura que permita acceder al aplicativo y no realizar ninguna actividad dentro del mismo.

Ilustración 15: Usuario tipo 4 - Usuario de Aplicativo

Este usuario por tener el privilegio solo de lectura dispone de la característica de Buscar es decir presentamos una opción en el que se

presenta un listado de todos los usuarios de aplicativo y poder realizar una búsqueda por distintos parámetros.



Ilustración 16: Buscar Usuario tipo 4 - Usuario de Aplicativo

## Reportes

Los reportes que se entregan como respaldo de la generación de los usuarios tipo 2 y tipo 3 son Actas de entrega/recepción y se los realiza por dos motivos:

- 1.- El acta se lo descarga en Formato Word editable, por si es necesario realizar algún acotamiento adicional, y es entregado al solicitante como respaldo del proceso realizado.
- 2.- Se realiza una impresión del acta para respaldo y auditoria futura el cual respalda la creación del usuario y es guardado en una caja fuerte física que se encuentra en custodia del área de seguridad.



Ilustración 17: Reporte - Acta de entrega/recepción



Para finalizar el proceso de la creación del usuario se genera el Acta de entrega de clave que permite a la persona o entidad solicitante disponer de la misma de manera física.



Ilustración 18: Acta de Entrega de Clave

### 3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

#### 3.1 Resultados

Los procesos que se realizan para validar que el sistema bancario es funcional y cumple con el objetivo de esta se describe a continuación en guiones de pruebas para cada flujo.

##### Módulo Usuario Caja Fuerte

###### a. Caso de prueba 1

**Caso de prueba:** Registro de usuario tipo 2

**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Crear un usuario nuevo

**Canal:** Aplicativo de Escritorio

**Estado:** Exitoso

**Evidencia:**

The screenshot shows a web application interface for user registration. The title is 'REGISTRO USUARIOS CAJA FUERTE'. The form is organized into several sections:

- General Information:** Includes fields for 'CÓDIGO' (00000001), 'USUARIO' (with a dropdown for 'ent'), 'NOMBRE DEL PROYECTO' (PRY - SOBROS\_PAGOS), 'SOLICITANTE' (VALLEJO MEZA, HENRY RAUL), 'CREADO POR' (ARCOS LAJANA), and 'DDI' (000000001).
- Classification and Environment:** Includes 'TIPO DE USUARIO' (LOCAL), 'AMBIENTE' (TEST), 'EMPRESA' (PROVEDOR), 'APLICACIÓN' (Autobent Serf DCA), and 'PLATAFORMA' (WINDOWS).
- Server and Production Details:** Includes 'NOMBRE DEL SERVIDOR' (ECBRTSFB1), 'IP DEL SERVIDOR' (172.31.2.6), 'RESPONSABLE XXX' (Rodríguez Barrera Luis Roberto), 'CUSTODIO XXX' (ADMINISTRACIÓN ENTIDADES), and 'OBSERVACIÓN' (Creación de usuario y clave para el DCA).
- Production and Justification:** Includes 'ESTADO' (ACTIVO), 'VIGENCIA' (PERMANENTE), 'ENTREGA SOBRE RESPONSABLE' (checkbox), 'REGISTRO CENTRALIZADA' (checkbox), 'APLICACIÓN PRODUCCIÓN' (checkbox), 'CLAVE' (checkbox), 'ACTA ENTREGA/RECEPCIÓN' (checkbox), and 'JUSTIFICACIÓN TÉCNICA' (Uso de las credenciales para el DCA).

At the bottom of the form, there is a 'GENERAR CONTRASEÑA' button with the value '5EAYEDuG7Z'. Below the form, there are two buttons: 'NUEVO' and 'GUARDAR'. A small note at the bottom left states: '\* Todos los campos son obligatorios, por favor no dejar espacios en blanco colocar la palabra APLICA'. A small note at the bottom right states: 'IMPORTANTE: En el campo observación se debe colocar la justificación técnica del usuario. Este campo es obligatorio y debe colocarse toda la información relevante del usuario.'

Ilustración 19: Registro de usuario tipo 2

En la ilustración 21 se procede a ingresar todos los datos solicitados para poder registrar al usuario tipo 2, es necesario tener en cuenta que el código es Autoincrementable y las "XXX" son el identificativo de la entidad financiera, para colocar el nombre del usuario se debe considerar que las 4 primeras letras representan el ambiente para el cual sera usado el usuario (Test, Producción o Desarrollo).

Adicionalmente tener en cuenta que en la observación es necesario colocar el justificativo de la creación del usuario tipo 2, de igual manera colocar el justificativo técnico que es muy importante.

Luego de haber llenado todos los campos procedemos a dar click en guardar.

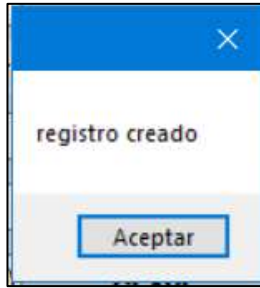


Ilustración 20: Registro ingresado exitosamente

La ilustración 22 informa que la información ingresada fue guardada de manera exitosa en la Base de datos.

## b. Caso de prueba 2

**Caso de prueba:** Consultar usuario

**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Consulta de usuario por Código

**Canal:** Aplicativo de Escritorio

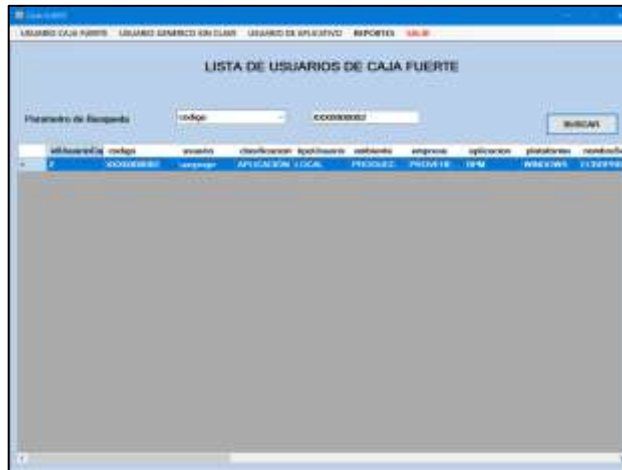
**Estado:** Exitoso

**Evidencia:**

Usuario/Código	código	usuario	clasificación	tipo/usuario	ambiente	empresa	aplicación	plataforma	control/rol
1	30000000	usufuete	APLICACIÓN LOCAL	TELE	PROVEER	Andover	WINDOWS	CCSISTEM	
2	30000000	usufuete	APLICACIÓN LOCAL	PRODUCC	PROVEER	3FM	WINDOWS	CCSISTEM	
3	30000000	usufuete	APLICACIÓN LOCAL	DESARRO	PROVEER	Directorio	WINDOWS	SGHINSOD	

Ilustración 21: Lista de usuarios tipo 2

En la ilustración 23 tenemos una lista del top 100 de usuarios de tipo 2, esta lista se presenta al momento de dar click en el sub modulo de usuario caja fuerte.



*Ilustración 22: Consulta de usuario tipo 2 por el parámetro de código*

En la ilustración 24 podemos observar la búsqueda realizada bajo el parametro de código.

### **c. Caso de prueba 3**

**Caso de prueba:** Consultar usuario

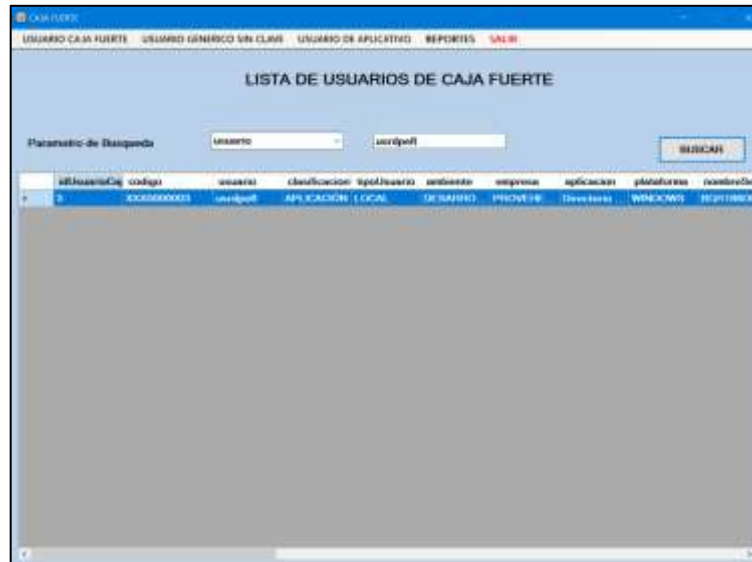
**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Consulta por usuario

**Canal:** Aplicativo de Escritorio

**Estado:** Exitoso

**Evidencia:**



*Ilustración 23: Consulta de usuario tipo 2 por el parámetro usuario*

Se puede realizar la búsqueda de usuario tipo 2 por medio de varios parámetros y uno de ellos el que se ve en la ilustración 25 que se lo realiza mediante el parámetro usuario.

#### **d. Caso de prueba 4**

**Caso de prueba:** Actualizar usuario

**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Actualizar el estado y la observación.

**Canal:** Aplicativo de Escritorio

**Estado:** Exitoso

**Evidencia:**

**ACTUALIZAR USUARIO CAJA FUERTE**

<b>CÓDIGO</b>	ANSW001	<b>NOMBRE DEL PROYECTO</b>	FFICORRE_FIGDS
<b>USUARIO</b>	admin	<b>SOLICITANTE</b>	VALLEJO PEZA HENRY RAUL
<b>CLASIFICACIÓN DE USUARIO</b>	APLICACION	<b>CREADO POR</b>	ARCOE OLIVERA
<b>TIPO DE USUARIO</b>	LOCAL	<b>ODIT</b>	
<b>AMBIENTE</b>	TEST	<b>FECHA DE CREACIÓN</b>	25-8-2023
<b>EMPRESA</b>	PROVEEDOR	<b>ESTADO</b>	INACTIVO
<b>APLICACIÓN</b>	Adicional Soft DCA	<b>VIGENCIA</b>	PERMANENTE
<b>PLATAFORMA</b>	WINDOWS	<b>ENTREGA SOBRE RESPONSABILIDAD</b>	
<b>NOMBRE DEL SERVIDOR</b>	SEBTP001	<b>REGISTRO CENTRALIZADA</b>	
<b>IP DEL SERVIDOR</b>	172.17.1.24	<b>APLICACIÓN PRODUCCIÓN</b>	
<b>RESPONSABLE XXX</b>	J. Rodriguez Gomez Luis Roberto	<b>CLAVE</b>	
<b>CUSTODIO XXX</b>	ADMINISTRACION ENTORNO	<b>ACTA ENTREGA/RECEPCIÓN</b>	
<b>OBSERVACIÓN</b>	Creación de usuario y clave para el DCA, por solicitud de banco	<b>JUSTIFICACIÓN TÉCNICA</b>	Uso de los credenciales para el DCA
<b>GENERAR CONTRASEÑA</b>	W#YD#G#2		

**ACTUALIZAR**

Ilustración 24: Actualizando el estado del usuario tipo 2 Inactivo.

Por regla de negocio y los usuarios tipo 2 creados solo pueden ser modificados ciertos parámetros es decir en la parte de actualización de usuario tipo 2, solo se toma en cuenta el Estado, responsable de banco y la observación por el cual se realiza la Actualización.

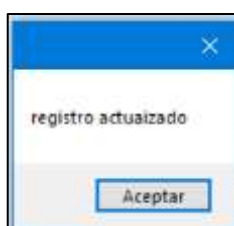


Ilustración 25: Registro Actualizado exitosamente.

Luego de haber modificado los parametros no nombrados damos click en actualizar para poder guardar la nueva información en Base de Datos y poder finalizar de manera exitosa.

## Módulo Usuario Genérico sin clave

### a. Caso de prueba 1

**Caso de prueba:** Registro de usuario tipo 3

**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Crear un usuario nuevo

**Canal:** Aplicativo de Escritorio

**Estado:** Exitoso

**Evidencia:**

CÓDIGO	0000200000	NOMBRE DEL PROYECTO	PRI
USUARIO	user	SOLICITANTE	Challana Olivo, Edison Fabian
TIPO	aplic	CREADO POR	TECA DIEGO
TIPO DE USUARIO	LOCAL	OOI	000000001
AMBIENTE	TEST	FECHA DE CREACIÓN	domingo, 27 de agosto de
EMPRESA	XXX	ESTADO	ACTIVO
APLICACIÓN	SOLC000	VIGENCIA	TEMPORAL
PLATAFORMA	WINDOWS	CUSTODIO XXX	ADMINISTRACION ENTIDADES
NOMBRE DEL SERVIDOR	E0SR79FE	RESPONSABLE XXX	Challana Olivo, Edison Fabian
IP DEL SERVIDOR	172.20.40.15	JUSTIFICACIÓN	PRI_RPA_F
GRUPO SERVIDOR	SISTEMA_OPERATIVO		

*Ilustración 26: Registro de usuario tipo 3*

Para la creación de usuario tipo 3 es necesario ingresar todos los datos que se presentan en la ilustración 28, el código es Autoincrementable y las letras “XXX” son el identificativo de la entidad bancaria.

Tener en cuenta que en el apartado de Empresa se coloca el nombre de la entidad financiera.

Al tener digitado todos los datos damos clic en guardar y los datos se almacenarán en la base de datos correspondiente.

**REGISTRO USUARIO GENERICO SIN CLAVE**

CÓDIGO	XXXX0000001	NOMBRE DEL PROMOTOR	PRR - RPA_FI
USUARIO	user - giga	SOLICITANTE	Orlinda Olivo, Edison Fabian
TIPO	APLICACIÓN	CREADO POR	TEDA (DEGO)
TIPO DE USUARIO	LOCAL	DNI	000000001
AMBIENTE	TEST	FECHA DE CREACIÓN	domingo, 27 de agosto de -
EMPRESA	XXX	ESTADO	ACTIVO
APLICACIÓN	SEL_000	TEMPORAL	TEMPORAL
PLATAFORMA	WINDOWS	ADMINISTRACIÓN ENTIDADES	-
NOMBRE DEL SERVIDOR	ECBR7SP01	Orlinda Olivo, Edison Fabian	
IP DEL SERVIDOR	172.20.48.11	PRR_RPA_FI	
GRUPO SERVIDOR	:SISTEMA_OPERATIVO	JUSTIFICACIÓN	

\* Todas las campos son obligatorias, por favor no dejar espacios en blanco indicar la palabra NO APLICABLE.

IMPORTANTE: En el campo observaciones se debe colocar la justificación técnica del usuario. Este campo es obligatorio y debe colocarse toda la información relevante del usuario.

**NUEVO** **GUARDAR**

Ilustración 27: Registro ingresado con éxito

En la ilustración 29 tenemos el mensaje de registro creado es decir el proceso de creación de usuario tipo 3 fue exitoso.

**b. Caso de prueba 2**

**Caso de prueba:** Consultar usuario tipo 3

**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Consulta de usuario por código

**Canal:** Aplicativo de Escritorio

**Estado:** Exitoso

**Evidencia:**



LISTA DE USUARIOS GENERICOS SIN CLAVE

Parametro de Búsqueda:

idUsuarioGe	codigo	usuario	tipo	tipoUsuario	ambiente	empresa	aplicacion	plataforma	nombreServ
1	XXXXC00	usrtgaga	APLICACIÓN LOCAL	TEST	XXXX	XXXX	SigloXXI	WINDOWS	ECBR1SF01
2	XXXXC00	usmpetu	APLICACIÓN LOCAL	PRODUCC	XXXX	XXXX	SigloXXI	WINDOWS	ECBR1SF01
3	XXXXC00	usrdomas	APLICACIÓN LOCAL	DESARRO	XXXX	XXXX	Destroder	WINDOWS	ECBR1SF02

Ilustración 28: Lista de usuarios tipo 3

En la ilustración 30 tenemos una lista del top 100 de usuarios de tipo 3, esta lista se presenta al momento de dar clic en el sub módulo de usuario genérico sin clave.

LISTA DE USUARIOS GENERICOS SIN CLAVE

Parametro de Búsqueda:

idUsuarioGe	codigo	usuario	tipo	tipoUsuario	ambiente	empresa	aplicacion	plataforma	nombreServ
3	XXXXC000002	usmpetu	APLICACIÓN LOCAL	PRODUCC	XXXX	XXXX	SigloXXI	WINDOWS	ECBR1SF01

Ilustración 29: Búsqueda de usuario tipo 3 por parámetro de código

En la ilustración 31 podemos observar la búsqueda realizada bajo el parámetro de código.

### c. Caso de prueba 3

**Caso de prueba:** Consultar usuario tipo 3

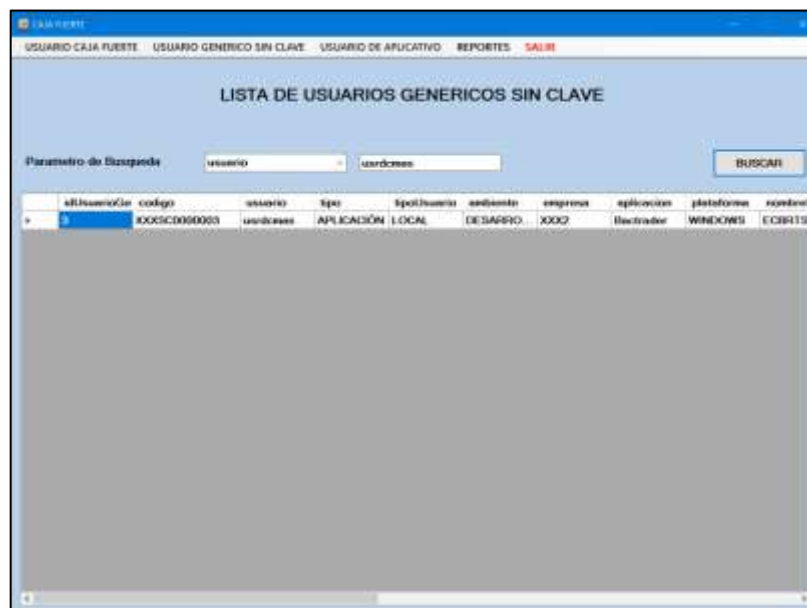
**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Consulta por usuario

**Canal:** Aplicativo de Escritorio

**Estado:** Exitoso

**Evidencia:**



*Ilustración 30: Consulta de usuario tipo 3 por el parámetro usuario*

Se puede realizar la búsqueda de usuario tipo 3 por medio de varios parámetros y uno de ellos el que se ve en la ilustración 32 que se lo realiza mediante el parámetro usuario.

### d. Caso de prueba 4

**Caso de prueba:** Actualizar usuario

**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Actualizar el estado y la observación.

**Canal:** Aplicativo de Escritorio

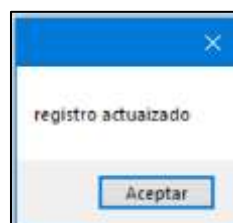
**Estado:** Exitoso

**Evidencia:**

The screenshot shows a desktop application window titled "ACTUALIZAR USUARIO GENERICO SIN CLAVE". The form is organized into several columns and rows of input fields and dropdown menus. The fields include: CODIGO (XXXXXXXXXX), USUARIO (usuario), TIPO (APLICACION), TIPO DE USUARIO (LOCAL), AMBIENTE (PRODUCCION), EMPRESA (Banco), APLICACION (Banco), PLATAFORMA, NOMBRE DEL SERVIDOR (ECONOMY), IP DEL SERVIDOR (172.25.48.13), GRUPO SERVIDOR (172.25.48.13), NOMBRE DEL PROYECTO, SOLICITANTE (Una Unidad - Head Accounting), CREADO POR (TELA DREDA), DOY (XXXXXXXXXX), FECHA DE CREACION, ESTADO (INACTIVO), VIGENCIA (TEMPORAL), CUSTODIO XXX (ADMINISTRACION CENTRALES), RESPONSABLE XXX (usuario Una Unidad (Jefe Regional)), and JUSTIFICACION (Gestionar Acciones de Registro Bank Un, pedido de banco). A large "ACTUALIZAR" button is positioned at the bottom center of the form.

*Ilustración 31: Actualización de estado del usuario tipo 3*

Por regla de negocio y los usuarios tipo 3 creados solo pueden ser modificados ciertos parámetros es decir en la parte de actualización de usuario tipo 3, solo se toma en cuenta el Estado, responsable de banco y la observación por el cual se realiza la Actualización.



*Ilustración 32: Usuario actualizado con éxito*

Luego de haber modificado los parámetros nombrados damos clic en actualizar para poder guardar la nueva información en Base de Datos y poder finalizar de manera exitosa.

## Módulo Usuario de Aplicativo

### a. Caso de prueba 1

**Caso de prueba:** Registro de usuario tipo 4


**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Crear un usuario nuevo

**Canal:** Aplicativo de Escritorio

**Estado:** Exitoso

**Evidencia:**



The screenshot shows a web application interface for user registration. The title bar indicates the current page is 'USUARIO DE APLICATIVO'. The main heading is 'REGISTRO DE USUARIOS DE APLICATIVO'. The form contains the following fields and values:

Field	Value
USUARIO	ADM
TIPO	SISTEMA OPERATIVO
SERVIDOR	SERVIDOR-SISTEMA-OPERATIVO
AUTENTICACION	Usuarios propios del Sistema operativo

Below the form, there are two buttons: 'NUEVO' and 'GUARDAR'. A note at the bottom of the form states: 'IMPORTANTE: En el campo observaciones se debe colocar la justificación técnica del usuario. Este campo es obligatorio y debe contener toda la información relevante del usuario.'

*Ilustración 33: Registro de usuario tipo 4*

Para los usuarios tipo 4 se solicita ingresar solamente ciertos datos los cuales son esenciales para mantener los registros y seguimientos necesarios para las auditorias.

Luego de colocar los datos solicitados damos clic en guardar para finalizar el registro del usuario tipo 4.

*Ilustración 34: Registro creado Exitosamente*

Al presentarse el mensaje de registro creado, esto significa que los datos fueron guardados en Base de Datos y finalizar el proceso.

**b. Caso de prueba 2**

**Caso de prueba:** Consultar usuario tipo 4

**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Consulta de usuario por usuario

**Canal:** Aplicativo de Escritorio

**Estado:** Exitoso

**Evidencia:**

The screenshot shows a web application window titled 'CAJA FUERTE'. The navigation bar includes 'USUARIO CAJA FUERTE', 'USUARIO GENÉRICO SIN CLAVE', 'USUARIO DE APLICATIVO', 'REPORTES', and 'SALIR'. The main heading is 'LISTA DE USUARIOS DE SISTEMAS'. Below the heading is a search section with 'Parámetro de Búsqueda' and a 'BUSCAR' button. The table below has the following data:

idUsuarioDeSistema	usuario	tipo	servidor	justificacion
1	ADM	SISTEMA OPERATIVO	ECRBEZ2-ECRBEZ2-ECRBEZ	Usuarios propios del Sistema s...
2	gencl	BASE DE DATOS	ECRBEZ10	El usuario gencl es un usuario...
3	gencl2	SISTEMA OPERATIVO	ECRBEZ21	Usuarios propios del Sistema s...

Ilustración 35: Lista de usuarios tipo 4 existentes

En la ilustración 37 tenemos una lista del top 100 de usuarios de tipo 4, esta lista se presenta al momento de dar click en el sub modulo de usuario de aplicativo.

The screenshot shows the same 'LISTA DE USUARIOS DE SISTEMAS' interface. The search section now has 'usuario' selected in the dropdown and 'gencl' entered in the search box. The table displays only one row, which is highlighted in blue:

idUsuarioDeSistema	usuario	tipo	servidor	justificacion
2	gencl	BASE DE DATOS	ECRBEZ10	El usuario gencl es un usuario...

Ilustración 36: Consulta de usuario tipo 4 por el parámetro usuario

Se puede realizar la búsqueda de usuario tipo 4 por medio de varios parámetros y uno de ellos el que se ve en la ilustración 38 que se lo realiza mediante el parámetro usuario.

### c. Caso de prueba 3

**Caso de prueba:** Consultar usuario tipo 4

**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Consulta por tipo

**Canal:** Aplicativo de Escritorio

**Estado:** Exitoso

**Evidencia:**



id Usuario De Sistema	usuario	tipo	servicio	publicacion
1	ADM	SISTEMA OPERATIVO	ECRINCO-ECRINO12-ECRIN	Usuarios propios del Sistema o...
3	ADM18	SISTEMA OPERATIVO	ECRINCO1	Usuarios propios del Sistema o...

*Ilustración 37: Consulta de usuario tipo 4 por el parámetro tipo*

Se puede realizar la búsqueda de usuario tipo 4 por medio de varios parámetros y uno de ellos el que se ve en la ilustración 39 que se lo realiza mediante el parámetro tipo.

## Módulo de Reportes

### a. Caso de prueba 1

**Caso de prueba:** Consulta y generación del Acta Recepción

**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Se entrega el documento de constancia de la solicitud de creación de usuario tipo 2, la consulta se realizará con el parámetro tipo de usuario.

**Canal:** Aplicativo de Escritorio

**Estado:** Exitoso

**Evidencia:**



*Ilustración 38: Generación del Acta de Entrega con los datos solicitados.*

Para poder tener un respaldo de cada uno de los usuarios creados se procede a la entrega del Acta Entrega Recepción.

La cual será entregado una copia al solicitante y la otra copia será respaldada en una caja fuerte de manera física.

Para poder obtener este reporte se realiza una consulta bajo el parámetro de búsqueda por usuario como se puede visualizar en la ilustración 40.

## **b. Caso de prueba 2**

**Caso de prueba:** Generación del documento de descarga

**Tipo de caso de prueba:** Positiva

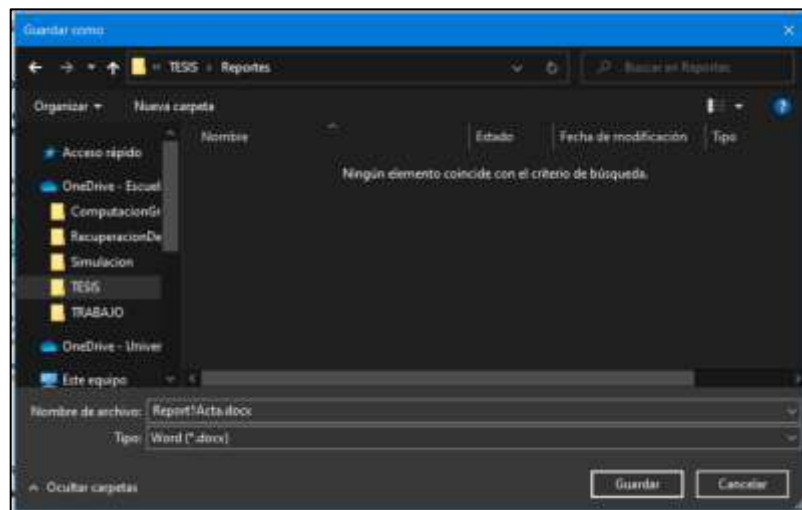


**Descripción del caso de prueba:** Generar el documento con extensión .docx para ser editado con datos adicionales necesarios en el acta de entrega.

**Canal:** Aplicativo de Escritorio

**Estado:** Exitoso

**Evidencia:**



*Ilustración 39: Guardar el reporte generado*

El reporte generado en la ilustración 40 nos permite descargarlo en formato pdf, excel y Word, en nuestro caso el que debemos guardar es el de extensión .docx para poder agregar datos adicionales y proceder a la entrega al solicitante y este flujo se lo visualiza en la ilustración 41.

**CONFIDENCIAL**

**COMPROBANTE DE FORMALIZACION DE ENTREGA DE USUARIOS Y CLAVES**

A la Sr. Nombre (cargo) portador de la cédula de identidad xxxxx, funcionario de la, le hacemos la entrega formal del usuario genérico detallado a continuación:

USUARIO	AMBIENTE	APLICACION	RESPONSABLE
usrtqar	TEST	Autoclient Swift DCA	Ingeniero II (Rodríguez Barrera Luis Roberto)

A través de este documento, me comprometo a mantener en un lugar seguro y bajo mi responsabilidad, los elementos que me han sido asignados. Así mismo, me comprometo a guardar la confidencialidad del documento. En caso de cambio de cargo o salida de la institución me comprometo a notificar vía mail a tiempo al Dpto de Administración de Claves y Riesgo Operativo el nombre del nuevo Responsable de la custodia. El incumplimiento de las responsabilidades asumidas en este documento, podrán estar sujetas a sanciones conforme al reglamento interno del Banco. Para constancia de la recepción del usuario indicado que desde la presente fecha xxxx están bajo su entera responsabilidad según lo especificado en el correspondiente certificado de confidencialidad que reposa en su carpeta personal en Recursos Humanos.

Firma: \_\_\_\_\_ Firma: \_\_\_\_\_  
Nombre: \_\_\_\_\_ Nombre: \_\_\_\_\_  
Cargo: \_\_\_\_\_ Cargo: \_\_\_\_\_  
Ci: \_\_\_\_\_ Ci: \_\_\_\_\_

*Ilustración 40: Reporte generado para ser editado y entregado.*

Una vez guardado el documento podemos acceder al mismo y agregar los datos que hacen falta como es el caso de Nombre e Identidad del solicitante.

**c. Caso de prueba 3**

**Caso de prueba:** Consulta y generación del Acta Recepción

**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Se entrega el documento de constancia de la solicitud de creación de usuario tipo 2, la consulta se realizará con el parámetro tipo de código.

**Canal:** Aplicativo de Escritorio

**Estado:** Exitoso

**Evidencia:**



Ilustración 41: Consulta del reporte bajo el parámetro tipo de código

Para obtener un reporte del Acta Entrega Recepción también se lo puede realizar por el parámetro de código como se visualiza en la ilustración 43 y obtendremos la información que se debe entregar al solicitante.

#### d. Caso de prueba 4

**Caso de prueba:** Acta de entrega de Clave

**Tipo de caso de prueba:** Positiva

**Descripción del caso de prueba:** Consulta del acta de entrega de Clave por el parámetro de usuario.

**Canal:** Aplicativo de Escritorio

**Estado:** Exitoso

**Evidencia:**



Ilustración 42: Consulta del Acta de entrega de clave

Como responsabilidad para las auditorías es necesario respaldar de forma física la clave generada para el usuario tipo 2, por lo que se genera un reporte bajo cualquiera de los siguientes parámetros usuario, tipo o código como se puede visualizar en la ilustración 44.

### 3.2 Conclusiones

- El uso de una herramienta como Caja Fuerte puede brindar un alto grado de confiabilidad al crear usuarios y contraseñas. Como utilidad de la encriptación de la clave del cliente, garantiza la seguridad de la información y reduce la enorme cantidad de acceso no autorizado. Además, como ser una aplicación de escritorio, es fácil de usar para aquellos que ya están familiarizados con el entorno de escritorio.
- Con base en el nivel de alta importancia al que debemos prestar atención al registrar claves, hemos llegado a la conclusión de que esta herramienta puede ofrecer un alto nivel de confiabilidad.

- Con el aplicativo Caja Fuerte, se brinda una experiencia de uso sencillo y se elimina la necesidad de recordar múltiples usuarios de distintos aplicativos.
- Gracias al progreso tecnológico a nivel global, se dispone de herramientas que facilitan el desarrollo de aplicaciones de forma rápida y segura, como es el caso del cifrado de datos.
- Hay una amplia gama de funciones de cifrado disponibles, las cuales deben ser evaluadas y algunas pruebas correspondientes para determinar su uso individual o combinado según los requisitos específicos.
- La idoneidad de los métodos y herramientas permite el desarrollo de aplicaciones para gestionar códigos privados de modo que se pueda preservar la privacidad, se pueda garantizar la confidencialidad.
- El enfoque iterativo-incremental para el desarrollo de aplicaciones ayuda significativamente en la gestión y seguimiento de avances funcionales, trabajar con algo sistemático ayuda mantener el flujo de trabajo bien definido.

### **3.3 Recomendaciones**

- Se recomienda para la base de datos, utilizar programación para encriptar los datos de registro, así el gestor de base de datos aunque sea manejada por un técnico, no sabría lo que contiene dicha base de datos y se mantendría la confidencialidad.
- Se recomienda implementar políticas de contraseñas, por ejemplo, no es recomendable utilizar frases, nombres o fechas comunes para la creación de contraseñas, por este motivo es mejor usar una forma aleatoria para mayor seguridad.

## 4 REFERENCIAS BIBLIOGRÁFICAS

- [1] José Fabián Roa Buendía, Seguridad Informática, McGraw-Hill/Interamericana de España, S. L. 2013.
- [2] Gabriel Baca Urbina, Introducción a la Seguridad Informática, Grupo Editorial Patria, Primera Edición EBOOK, México, 2016.
- [3] Alfonso García, Cervigón María, Seguridad Informática, Ediciones Paraninfo, SA 1ra edición, 2011.
- [4] José Antonio Gonzáles Seco, Lenguaje de programación C#, 2002.
- [5] Brice Arnaud Guérin, ASP.NET con C# en Visual Studio 2017.
- [6] Ángel Arias, Aprende a Programar ASP.NET y C#, 2da Edición.
- [7] Schwaber, K., & Sutherland, J. (2013). "Guía Scrum". Scrum.org.