Implementación de una aplicación móvil de mensajería con cifrado simétrico por sustitución y desplazamiento

Alcides Augusto Monterroza Quintero[1] Código 20201093005, José Alejandro Buitrago [1] Código 20201093002
1. Universidad Distrital Francisco José de Caldas – Especialización en Teleinformática
aamonterrozaq@correo.udistrital.edu.co, jabuitragot@correo.udistrital.edu.co

Resumen **–** En este trabajo se construye una experiencia académica que, mediante la implementación de una aplicación de mensajería cifrada, busca la comprensión de dos métodos clásicos de cifrado aplicados a una de las formas de comunicación más usadas en la actualidad, llevándolos al escenario de la programación y desarrollo de aplicaciones móviles. Para tal fin se escogieron los cifrados por sustitución y por desplazamiento, se hizo una relación de requerimientos de la aplicación, se identificaron y analizaron casos de uso, se definieron las clases y se realizó el desarrollo del código en JAVA con conexión a base de datos SQL en Azure utilizando servicios Web desarrollados en C#, e interfaz de usuario con el IDE Android Studio.

Palabras Clave **–** Aplicación móvil, Android, IDE Android, cifrado, nube, servicio web, base de datos, Lenguaje Java.

Abstract **-** In this work an academic experience is built that by implementing an encrypted messaging application, seeks to understand two classic encryption methods applied to one of the most used forms of communication today, taking them to the programming and development scenario of mobile applications. For this purpose, the substitution and displacement ciphers were chosen, a list of application requirements was made, use cases were identified and analyzed, classes were defined, and the code was developed in Java with an SQL database connection in Azure using web services developed in C# , and UI with the Android Studio IDE.

Keywords **–** Mobile application, Android, Android IDE, encryption, Cloud, web service, database, Java SDK.

I. INTRODUCCIÓN

El presente proyecto el cual es con intenciones académicas y de estudio, abordaremos áreas de gran interés en el campo de la informática. A través del tiempo, el valor y la importancia que ha ganado el hecho de tener y/o generar información, ha trascendido de forma tal que es indispensable poseer o tener acceso a la misma para tomar decisiones. La criptografía se ha encargado de los asuntos relacionados con el transporte y alojamiento seguro de la información y se ha hecho cada vez más relevante e importante en todos los aspectos que dan tratamiento a la misma, ya sea en lo personal como en todas las áreas de negocio que son de bastante importancia para conservar la confidencialidad y la legitimidad de la información.

Tratándose de la transmisión de la información, la mensajería es una de las formas que también a través del tiempo ha evolucionado y que se ha mantenido vigente por la necesidad indispensable de estar al tanto de los hechos que generan información y que, a la practicidad de hoy, se tiene tan rápido como las tecnologías lo permitan. Por ello es importante que la misma se mantenga y se transporte de manera segura, el cual es el objeto de este proyecto, mostrar la forma en que se puede transportar la misma de terminal a terminal sin ser inteligible para sujetos ajenos a los originalmente interesados.

II. OBJETIVOS

i. Objetivo General

Implementar una aplicación móvil de mensajería usando métodos clásicos de cifrado simétrico.

ii. Objetivos Específicos

a. Identificar los requerimientos funcionales y no funcionales para la aplicación móvil de mensajería.
b. Identificar y diseñar el diagrama de casos de uso de la aplicación móvil de mensajería.
c. Diseñar el modelo de datos de la aplicación móvil de mensajería.
d. Desarrollar los algoritmos de cifrado simétrico por sustitución y por desplazamiento en el lenguaje de programación Java.
e. Implementar un servicio API REST en lenguaje de programación C# para conectar a la base de datos de la aplicación móvil.
f. Implementar una base de datos en nube de Microsoft Azure basada en el servicio PaaS de SQL Server.
g. Diseñar e implementar la capa de presentación de la aplicación móvil de mensajería usando el IDE Android Studio.
h. Realizar las pruebas de funcionamiento de la aplicación móvil de mensajería.

III. PLANTEAMIENTO DEL PROBLEMA

El auge en el uso de las herramientas de tecnologías de la información y las comunicaciones en la cotidianidad y en el mundo laboral, trae consigo retos desde el punto de vista de la seguridad de la información que viaja a través de las redes, es por esto que la criptografía juega un papel fundamental a la hora de proteger la información.

Al tener contacto con estas herramientas tecnológicas, las personas están exponiendo sus datos de forma inevitable al riesgo de ser interceptados y ser accedidos por entidades o

personas distintas al destinatario. Es el caso de las herramientas de mensajería, donde se requiere contar con técnicas de cifrado que garanticen la confidencialidad de la información.

En general los consumidores de herramientas de mensajería desconocen la complejidad del proceso y andamiaje que existe detrás de éstas, y sólo se limitan a utilizarlas sin preguntarse el trato que se le está dando a su información en cuanto a almacenamiento y tránsito seguro, de igual forma se desconoce el funcionamiento de los algoritmos de cifrado. Además de lo anterior, en muchas herramientas móviles de mensajería, los mensajes al llegar al receptor son totalmente legibles, lo cual hace que, si un intruso tiene acceso físico al dispositivo móvil y a la aplicación, pueda leerlos fácilmente.

Con este trabajo se busca implementar una herramienta móvil de mensajería con funcionamiento básico, recurriendo a la aplicación de algoritmos de cifrados clásicos, de tal forma que estudiantes o público en general tengan una idea más clara del proceso realizado para dar confidencialidad a los datos enviados y tener una base para el desarrollo de futuros proyectos, o para que usuarios finales de estas herramientas puedan saber el grado de compromiso de la confidencialidad de su información.

IV. HIPÓTESIS

La implementación de una aplicación móvil de mensajería usando métodos clásicos de cifrado simétrico, servirá para poner en práctica conceptos de programación de algoritmos de cifrado, desarrollo de aplicaciones móviles, de redes y bases de datos como especialistas en Teleinformática, y proporcionará una base académica para el desarrollo de proyectos de criptografía e influirá en el uso más consciente de las herramientas móviles de mensajería por parte de usuarios finales.

V. ALCANCES Y LIMITACIONES

Definiremos con precisión los aspectos del panorama de alcances y limitaciones del producto de este proyecto, el cual como ya conocemos será la aplicación de mensajería cifrada de punta a punta. Para este caso, es importante aclarar que las limitaciones en sí no representan una complicación o una dificultad para el desarrollo del proyecto.

i. Alcances

a. El producto final abordará lo relacionado con mensajería de texto de usuario a usuario a través de un ID. Esto implicaría en que cada usuario manejaría su cuenta y podrá enviar mensajes a un usuario ya conocido.
b. El cifrado de mensajes se realizará en el momento en que se da en la opción “Send” y se desencripta al dar en la opción “Receive”, esto como parte de la fase experimental del proyecto de aplicación.
c. La aplicación podrá cifrar la información de manera embebida, lo cual significa que el algoritmo de encriptación estará dentro de la misma aplicación.
d. El producto final contará con una base de datos,

   en la que se evidenciarán los mensajes finales enviados en los 2 sentidos.
e. El proyecto hará énfasis en el desarrollo de dos algoritmos clásicos para cifrar mensajes generados por los usuarios de la aplicación móvil Android.

ii. Limitaciones:

a. Este proyecto de aplicación es una versión beta únicamente con fines académicos y experimentales, ya que en el mercado existen herramientas que hacen cifrado de punta a punta, con algoritmos poderosos.
b. La aplicación no soportará envío y recepción de archivos multimedia, únicamente texto plano con el que se pueda identificar la encriptación y el cifrado del mismo a través de herramientas de análisis de tráfico.
c. El desarrollo del proyecto estará enfocado a la parte funcional y no a la parte del diseño gráfico de la interfaz de la aplicación.

VI. JUSTIFICACION

En la actualidad, las comunicaciones a través de mensajería son indispensables para el cotidiano de esta generación y de todos sus componentes de vida. La mensajería instantánea ha facilitado la forma y la velocidad en cómo llegamos a los demás, y por supuesto nos permite tomar decisiones en plazos y tiempos en los que tal vez, jamás imaginamos. Sin embargo, muchas veces pasamos por alto que la mensajería no debe cumplir con las expectativas del envío y recepción de mensajes, sino que también deben ser de manera segura y fidedigna.

El presente trabajo no es una novedad, pero sí es una experiencia tecnológica que nos permite evidenciar los conocimientos académicos recopilados a lo largo de nuestra especialización y que permite abarcar las capas lógicas necesarias para que una herramienta de mensajería pueda tener una capa de seguridad como lo es, el cifrado de la información de punta a punta. Esto permitirá contribuir con el fortalecimiento y la experiencia académica propia y de otros estudiantes, al evidenciar que no solo hoy día ya se cuenta y existen estas herramientas en nuestro cotidiano, sino que tendremos la oportunidad de ver su planeación, desarrollo y funcionamiento de lo que consideramos, es transparente para la experiencia del usuario.

El realizar este proyecto de mensajería cifrada puede representar un impacto realmente interesante en la comunidad a la que pueda llegar esta experiencia como se describe a continuación: en su formación académica, en cómo vemos y percibimos las herramientas tecnológicas, ya no en su forma, sino que en su fondo a través del conocimiento de su funcionamiento y hará conciencia en cómo usamos las herramientas de mensajería en el cotidiano y finalmente en cómo podemos construir a partir de la

abstracción de una necesidad específica, pero de manera segura y confiable.

VII. MARCO TEORICO

- **Web Service**

Hablando de sistemas informáticos de comunicación e interoperabilidad, los webs Service son propiamente este tipo de soluciones que otorgan flexibilidad a la hora de integrar diferentes sistemas y comunicarlos de manera transparente.

Los servicios web están configurados para realizar un conjunto de tareas específicas el cual expone directamente a sistemas interesados dichas funciones, las cuales son consumidas a petición, entregado o recibiendo información en red entre los sistemas de información, lo que significa que un servicio web del tipo que sea, está en la capacidad de ser la interface de comunicación entre 2 sistemas que posiblemente hablen diferente idioma pero que pueden llegar a tener intereses en particular cumpliendo con los protocolos de comunicación en la red. [1]

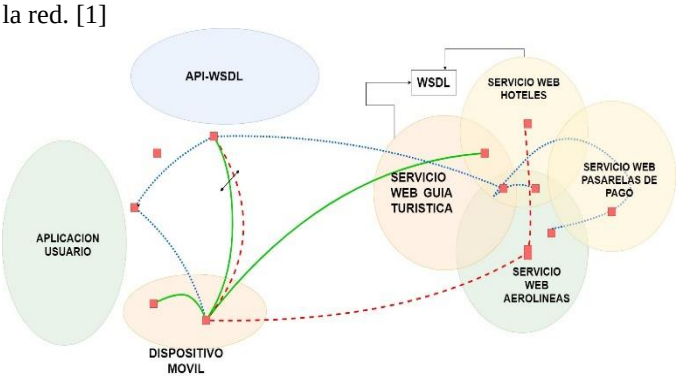


Figura 1. Flujo peticiones y respuestas a un servicio web.

Un escenario típico de servicios web es una aplicación de empresa que solicita un servicio de otra aplicación existente. La solicitud se procesa con una dirección web determinada, utilizando mensajes SOAP a través de un transporte HTTP, JMS (Java Message Service) o se invoca directamente como EJB (Enterprise JavaBeans). El servicio recibe la solicitud, la procesa y devuelve una respuesta. Ejemplos de un servicio web pueden ser los informes meteorológicos o la obtención de información bursátil.

Características:

a. Utilización de estándares de internet. La única forma para que los servicios Web sean utilizados por la cantidad de sistemas heterogéneos existentes en Internet es el empleo del protocolo de transferencia de datos HTTP utilizado por todos los navegadores Web y XML.

b. Basados en tecnologías de paso de mensajes. La interacción entre el cliente y el proveedor del servicio es empaquetada en unidades autodescriptivas denominadas mensajes. Dicha interacción se describe en función de los mensajes intercambiados.

a. Combinan lo mejor de la tecnología de componentes y de la tecnología Web. Los servicios Web presentan una funcionalidad de caja negra que puede ser reutilizada sin preocuparse de cómo es implementada y ello proporciona interfaces bien definidas.

- **Aplicación Móvil**

Las aplicaciones móviles, al igual que cualquier otra aplicación, son algoritmos definidos que obedecen a instrucciones ejecutadas en una instancia de cómputo arrojando unos resultados esperados a partir de sentencias de entrada. A diferencia de las aplicaciones convencionales, las aplicaciones móviles son aplicaciones informáticas diseñadas para ser ejecutadas en teléfonos inteligentes, tabletas y otros dispositivos que impliquen movilidad. Cuando hablamos de movilidad, las aplicaciones permiten a los usuarios efectuar conjunto variado de tareas profesionales, de ocio y educativas que facilitan cualquier actividad del diario vivir.

Hoy día existen diversas plataformas que las soportan, es decir, sistemas operativos de diferentes fabricantes que son el núcleo para que estas funcionen y puedan ser interoperadas en los diferentes dispositivos móviles donde sean instaladas, pudiendo hacer uso de los recursos disponibles como, sensores, cámara, luces, gps, etc. el sistema operativo móvil más conocido es Android, el cual funciona en millones de dispositivos móviles en el mundo y este a su vez, soporta millones de aplicaciones existentes en su tienda.

Existen aplicaciones móviles gratuitas y otras de pago, donde en promedio el 20 a 30 % del coste de la aplicación se destina al distribuidor y el resto es para el desarrollador. El término app se volvió popular rápidamente, tanto que en 2010 fue listada por la American Dialect Society como la palabra del año. [2]

Al ser aplicaciones residentes en los dispositivos están escritas en algún lenguaje de programación compilado, y su funcionamiento y recursos se encaminan a aportar una serie de ventajas [2] tales como:

a. Accesos eficientes y de primera mano, tratándose de movilidad, la información está disponible en todo momento.
b. Almacenamiento de datos personales de manera segura.
c. Muy práctico con respecto a aplicaciones de tipo PC.
d. Conectividad más eficiente y alta disponibilidad de servicios a través de las mismas aplicaciones.

Muchas de las aplicaciones móviles basadas en servicios, son plantillas que traen información a la capa de presentación, es decir, se instala formatos base desarrollados en un lenguaje de programación y que, a través de conexión a internet, consultan y consumen servicios web los cuales proveen la información para que estos sean presentados en la aplicación. Esta es una de las formas más básicas de crear aplicaciones

móviles, las cuales son basadas en su mayoría, en servicios web. Otros tipos de aplicaciones que al igual requieren conectividad, consumen más recursos puesto que la exigencia gráfica, de procesamiento y almacenamiento son necesarias para su correcto funcionamiento, por ejemplo, los juegos, los cuales son desarrollados en un lenguaje de programación compatible y que al ser instalados consumen servicios para traer información necesaria y complementar el funcionamiento de la aplicación.

- **Mensajería Electrónica**

La mensajería electrónica es básicamente el sistema de intercambio de mensajes sin un soporte papel [3], lo que significa que es de ordenador a ordenador, o para muchos de los casos, de sistema a sistema. La mensajería electrónica puede transmitirse en cualquier sistema de red de comunicaciones siempre y cuando se cumpla con determinados protocolos.

1. Cifrado de información.

En el mundo de la informática, el cifrado es la conversión de datos de un formato legible a un formato codificado, que solo se pueden leer o procesar después de haberlos descifrado [4], básicamente, significa, aplicar un algoritmo de cifrado determinado junto con una clave, a una determinada información que se requiere trasmitir y no ser inteligible a la interpretación de un sistema o persona.

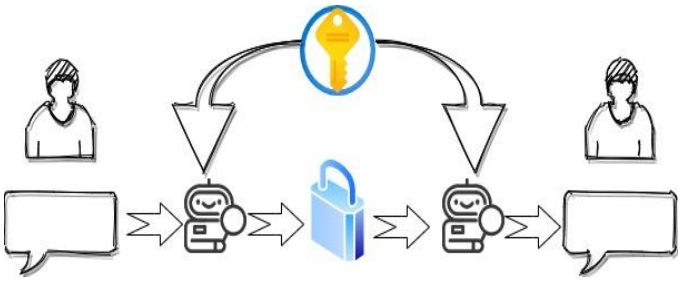


Figura 2. Flujo cifrado y descifrado de un mensaje.

El cifrado es el elemento fundamental de la seguridad de datos y es la forma más simple e importante de impedir que alguien robe o lea la información de un sistema informático con fines malintencionados. Utilizado tanto por usuarios individuales como por grandes corporaciones, el cifrado se usa ampliamente en Internet para garantizar la inviolabilidad de la información personal enviada entre navegadores y servidores. Esa información podría incluir todo, desde datos de pagos hasta información personal. Las empresas de todos los tamaños suelen utilizar el cifrado para proteger los datos confidenciales en sus servidores y bases de datos. [4]

2. ¿Por qué cifrar la información?

Teniendo la ventaja más obvia de proteger la información contra la lectura ajena o algún tipo de amenaza que no pueda ser contenida por un sistema de seguridad, las diferentes técnicas de cifrado ofrecen un medio para demostrar la autenticidad y el origen de la información. Con lo anterior, los métodos de cifrado proveen características naturales para poder verificar el origen de un mensaje y así confirmar que no ha sufrido ningún tipo de modificación durante el momento de la transmisión.

3. La llave de entrada

Lo esencial del cifrado es el concepto de algoritmos de cifrado y "claves". Cuando se envía información, esta se cifra mediante un algoritmo y solo se puede descodificar con la clave apropiada. Una clave puede almacenarse en el sistema de recepción, o bien, transmitirse junto con los datos cifrados.

4. Métodos de Cifrado

Para cualquier técnica aplicada en el tipo de cifrado se utilizan una serie de métodos para cifrar y descifrar la información, y estos métodos van evolucionando a medida que las tecnologías van cambiando y desarrollando formas en cómo se ingresa a los sistemas y se intercepte la información. [5]

a. Cifrado de clave simétrica: también conocido como algoritmo de clave secreta, es un singular método de descodificación de mensajes que debe ser provisto al receptor antes de que el mensaje se pueda descodificar. La clave que se usa en la codificación es la misma que se utiliza en la descodificación, lo que resulta más conveniente para los usuarios individuales y los sistemas cerrados. De lo contrario, se le tiene que enviar la clave al receptor, lo que aumenta el riesgo de alteraciones en caso de que terceros, como un hacker, la intercepten. La ventaja es que este método es mucho más ágil que el método asimétrico.
b. Cifrado de clave asimétrica: este método utiliza dos claves diferentes (pública y privada), que están vinculadas entre sí matemáticamente. Las claves son solo números extensos vinculados entre sí, pero no son idénticos, de ahí el término asimétrico. La clave pública se puede compartir con cualquier persona, mientras que la clave privada debe mantenerse en secreto. Ambas se pueden usar para cifrar un mensaje, y la clave opuesta a la que se emplee para cifrarlo se utiliza luego para descodificarlo



Figura 3. Métodos de cifrado.

Para el caso de estudio presente, tenemos dos tipos de técnicas de cifrado simétrico como lo es, sustitución y desplazamiento o técnicamente llamado cifrado Atbash y cifrado cesar, respectivamente.

a. Cifrado Atbash (sustitución): el cifrado por sustitución consiste básicamente en reemplazar un carácter conocido por otro. Para lo cual se conforman unas tablas en las que se especifica un carácter original el cual es reemplazado o sustituido por otro carácter llamado clave.
b. Cifrado Cesar (desplazamiento): el cifrado por desplazamiento consiste en desplazar un numero establecido de caracteres en un orden especifico, esto quiere decir que al tener todos los tipos de caracteres originales; letras, números, símbolos como un patrón para definir el desplazamiento, fijo un numero de caracteres a desplazarme y este carácter al cual me desplace, será mi carácter clave. De esta manera voy reemplazando mi mensaje original, por mi mensaje cifrado. [7]

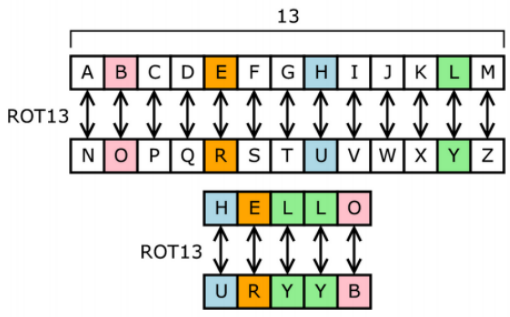


Figura 4. Cifrado por desplazamiento del alfabeto por 13 posiciones. [7]

VIII. PROCESO DE DESARROLLO

Inicialmente trabajaremos en todo el proceso necesario para construir y desarrollar software, el cual radica en establecer diagramas en donde interactúan actores, sistemas, entidades, software de core, dispositivos, etc. posteriormente, establecemos el framework, lenguaje y por supuesto el IDE de programación y diseño Android Studio.

Como bien sabemos es un caso de estudio, en el que desarrollaremos una aplicación funcional en lenguaje JAVA y su parte de diseño y capa de presentación en la versión 4 de Android Studio, allí se integrara en una interfaz única todo el desarrollo de nuestra aplicación, además de contar con lo necesario para comunicarse con un API desarrollada en C# la cual contiene los métodos necesarios para procesar las peticiones de la aplicación y finalmente insertarlas en una base de datos que está en la nube. A continuación, todo el proceso estructural del cual hemos hecho énfasis en esta parte inicial del proceso de desarrollo.

- **Relación Básica de Requerimientos**

Nombre de la ciudad	BOGOTÁ
Información recopilada por	ALCIDES MONTERROZA Y JOSÉ ALEJANDRO BUITRAGO
Responsables	ADMINISTRADOR DE LA APP

NÚMERO	FECHA	DESCRIPCIÓN	FUNCIONARIO
01		ADMINISTRAR INFORMACIÓN DE USUARIOS REGISTRADOS A TRAVÉS DE LA APLICACIÓN.	ADMINISTRADOR
02		EL USUARIO DEBE PODER REGISTRARSE EN LA APP, CREAR,CONSULTAR Y ENVIAR MENSAJES DE TEXTO A UN DESTINATARIO QUE TAMBIÉN SE ENCUENTRE REGISTRADO EN LA APP.	USUARIO
03		EL SISTEMA DEBE CIFRAR LOS MENSAJES DE TEXTO POR EL MÉTODO DE CIFRADO CLÁSICO DE DESPLAZAMIENTO Y CIFRAR LA LLAVE POR EL MÉTODO DE SUSTITUCIÓN.	SISTEMA
04		EL SISTEMA DEBE ALMACENAR EN UNA BASE DATOS DE LA NUBE DE AZURE LOS MENSAJES CIFRADOS.	SISTEMA

- **Listado de Actores**

NÚMERO	ACTOR (ROL)	USUARIO	COMENTARIO O DES.
01	ADMINISTRADOR	ADMINISTRADOR DE LA APLICACIÓN	
02	USUARIO	USUARIO DE LA APP	
03	SISTEMA	LÓGICA DE LA APLICACIÓN	

- **Diagrama de casos de uso**

En el diagrama de casos de uso (figura 5) vemos los distintos actores, incluyendo el sistema, ya que es necesario representarlo por conectar casos importantes para que se exprese en su totalidad la funcionalidad de la aplicación. Contamos con un usuario administrador que cumplirá con funciones de base como la modificación y eliminación de usuarios, en estrictos y condicionales circunstancias, sin pasar de alto que este debe hacer previo inicio de sesión en la misma APP, la cual listara los usuarios que hacen parte del sistema.

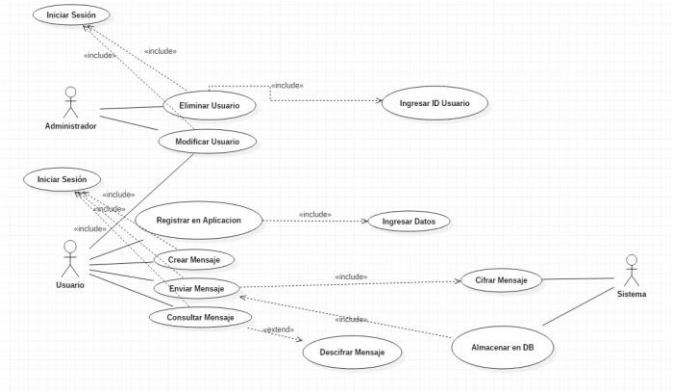


Figura 5. Diagrama de casos de uso

El usuario es el actor más representativo por la naturaleza de su rol, ya que es el que necesariamente realizara interacciones funcionales de la aplicación en sí. El usuario estará en la capacidad de hacer un registro de tipo *first time* para posteriormente iniciar sesión, de lo que desplegará un menú en el que podrá crear un mensaje, enviarlo y consultar un mensaje en su buzón de mensajes por leer, no sin antes descifrarlo. El sistema cumple un rol implícito en su condición de caja negra, está en la capacidad de cifrar un mensaje antes de enviarlo y almacenarlo en una base de datos. Es así, como vemos en el

diagrama de casos la representación de interacción entre actores que estarán presentes en nuestra aplicación.

- **Diagrama de clases**

De acuerdo con el diseño y funcionalidad de la aplicación, se modela el diagrama de clases (**figura 6),** en el que encontramos una relación entre dos clases conocidas como Usuario y Mensaje. En ellos existe una multiplicidad dada, unos atributos localizados en cada una de las clases y por supuesto los métodos que ejecutan funciones a partir de instrucciones dadas para que la aplicación pueda ser funcional. Como se pudo ver anteriormente, el modelado y diseño de este diagrama de clases, nos indicara a partir de la abstracción lo que debemos programar, los objetos, los métodos y la interacción entre cada una de esas clases y métodos.

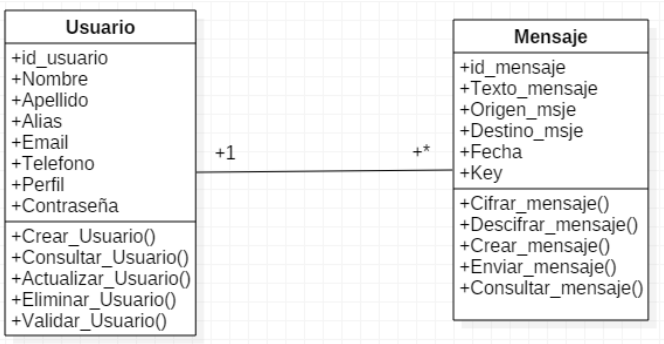


Figura 6. Diagrama de Clases.

Vimos entonces en la **Figura 6** como y de qué forma están relacionadas las clases, en la que se estipula una bidireccionalidad en que por defecto las clases acuden una a la otra o mejor aún, la una es consciente de la otra para poder hacer la interacción. En la figura 6 vemos entonces, como la clase Usuario puede tener una o más instancias de mensajes asociados a ella.

- **Diagrama de despliegue**

De acuerdo con toda la lógica de la implementación, y como una de las formas más sencillas de representar el flujo bidireccional, componentes y funcionamiento de la aplicación en lo que llamamos caja negra, se diseñó el diagrama de despliegue (figura 7). En este caso, el diagrama de despliegue representa la implementación física de todo el flujo de información generada por la aplicación (Mensajería Cifrada) y los diferentes componentes como *APIs,* bases de datos y la manera en cómo las interfaces ejecutan un claro ejemplo de interoperabilidad entre cada una de las cajas contenedoras.

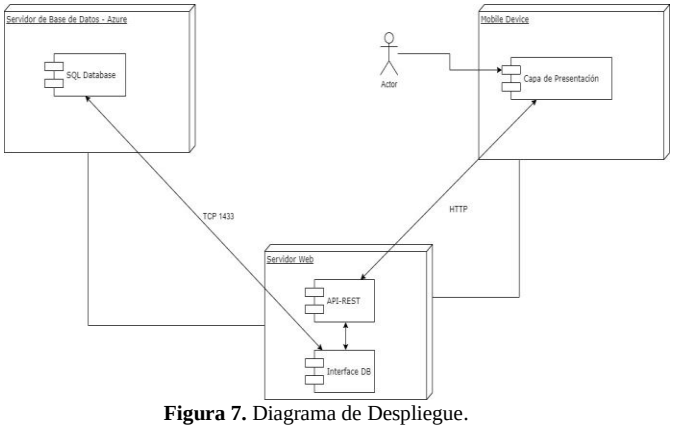


Figura 7. Diagrama de Despliegue.

En el diagrama de despliegue (figura 7), encontramos una serie de protocolos representados entre los componentes y como se hacen necesarios para que el artefacto (información generada) transite de manera tal que pueda cumplir con la funcionalidad planteada. Por último se representa la relación entre cada uno de los nodos y el orden en que se conectan entre sí, representados por flechas en ambas direcciones que fácilmente indican el flujo de información entre los nodos.

IX.RESULTADOS

- **Registro y Login.**

Al iniciar la aplicación *"Mensajería Cifrada"* encontramos una interfaz inicial que nos permite 2 opciones, en primera instancia ingresar a la misma de existir un usuario, o de lo contrario si no se cuenta con este, permite iniciar el registro como se aprecia en el conjunto de figuras a continuación (figura 8).

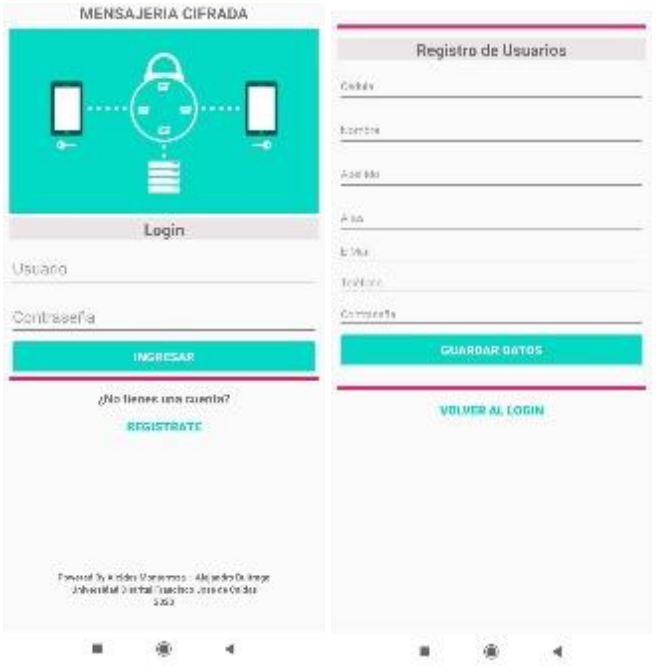


Figura 8. Interfaz de Autenticación y Registro.

- **Inicio y Bandeja de entrada**

Posterior al inicio de sesión o autenticación del usuario, es importante mostrar el contenido de su bandeja de entrada, allí encontramos nuevamente el nombre de la aplicación, el nombre de la especialización, el usuario, logo de la Universidad Distrital y por supuesto un menú de opciones, el cual veremos en el siguiente paso de resultado.
Al explorar la bandeja que llamamos *"Mis Mensajes"* encontramos los mensajes a ser leídos, encontrando 2 indicadores de mensajes *"leídos" y "No leídos"* con ello tenemos la certeza de que mensaje procesar de acuerdo con su estado. Al seleccionar uno de los mensajes, es necesarios descifrar la llave con la que viene cifrado el mensaje, de lo contrario no es posible ver el mensaje. Posterior a ello, desciframos el mensaje, el cual se muestra cifrado en la bandeja, pero descifrado en la parte inferior como se muestra en la figura 9 con el mensaje *"hola inge"*. La opción *"Actualizar bandeja"* permite actualizar para visualizar nuevos mensajes, sin embargo, cada vez que llegue un nuevo mensaje tendremos alertas de tipo *push* (figura 10) para estar al tanto de nuestra bandeja de entrada.

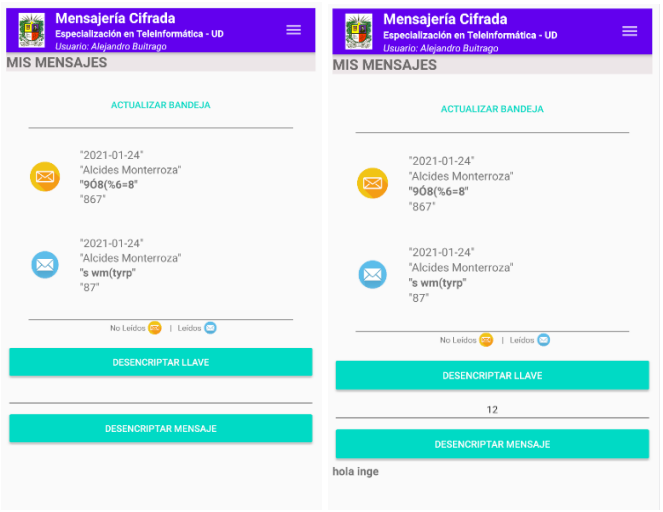


Figura 9. Interfaz de Inicio y Bandeja de Entrada.

Las notificaciones de tipo *push* basadas en el servicio *firebase* de Google son bastante importantes a la hora de crear aplicaciones móviles, ya que allí genera una alerta cuando existe alguna novedad en la misma (figura 10). Al pulsar allí, de inmediato nos lleva al inicio o al login, de ser necesario.

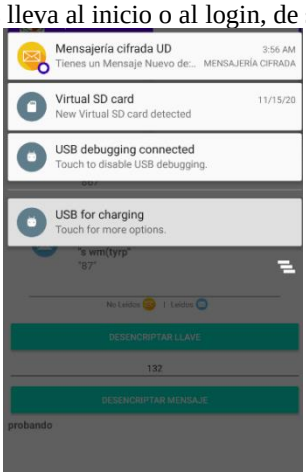


Figura 10. Interfaz de Inicio y Bandeja de Entrada.

- **Menú de Opciones usuario**

Sobre el mismo inicio, tenemos la cinta de menú que está ubicada en la parte superior derecha de nuestra pantalla. Allí encontraremos 4 opciones disponibles para interactuar en la aplicación de *"Mensajería Cifrada"* (figura 11). Veremos cada una de ellas a continuación.

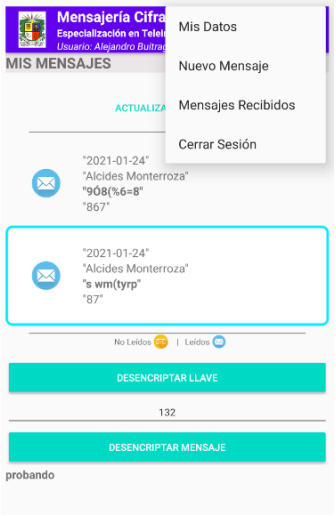


Figura 11. Menú de Opciones.

1. Mis Datos

En esta opción encontramos los datos que generó cada usuario en el registro, por lo que es posible modificarlos excepto, la cedula. Esta última es necesaria para identificar al usuario y poder modificarla crearía una inconsistencia relacional en la base de datos. Adicionalmente, encontramos la opción de *"Darse de Baja"* ya que allí es posible eliminar su usuario del sistema sin quedar rastro de este.



Figura 12. Opción del Menú Mis Datos.

2. Nuevo Mensaje

En esta opción encontramos nuestra lista de contactos disponible para redactar un mensaje a alguno de ellos. Se encuentra la fecha actual en la que se redactara un mensaje y podemos hacer uso del buscador en caso de que nuestra lista sea amplia. Posterior al elegir un contacto procedemos a redactar un mensaje y generaremos una llave privada, la cual será necesaria para cifrar el mensaje y descifrarlo en el destino. Como se muestra en la figura 13, vemos que al enviar al mensaje se genera una notificación de tipo *Toast* la cual nos indica que el mensaje se envió correctamente.

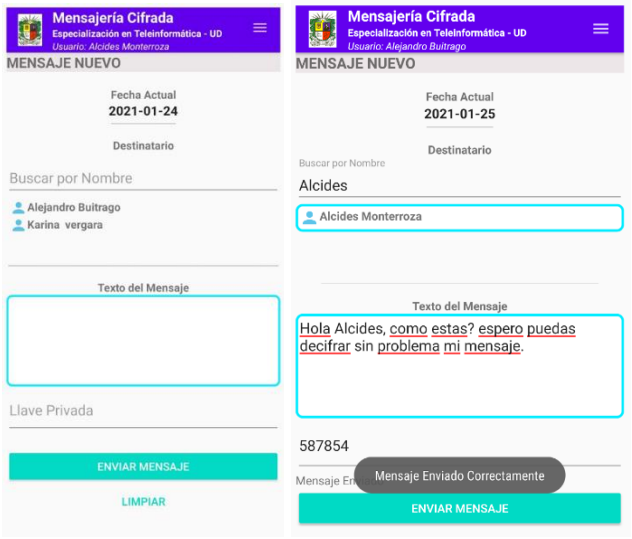


Figura 13. Opción del Menú Nuevo Mensaje.

3. Mensajes recibidos

En esta opción simplemente nos regresa al inicio o Bandeja de Entrada como se muestra en la figura 9.

4. Cerrar sesión

En esta última opción del menú, cierra la sesión existente destruye la conexión y nos carga nuevamente la interfaz de Registro y Login como se muestra en la figura 14.



Figura 14. Opción del Menú Administrador.

5. Administración

Esta opción solo está presente en aquellos usuarios que tienen perfil de administrador, y que tienen la capacidad de hacer cambios en los usuarios a través del id del mismo. Adicionalmente, y bastante importante, este puede cambiar el tipo de perfil de usuario normal, a usuario administrador como se muestra en la figura 15.

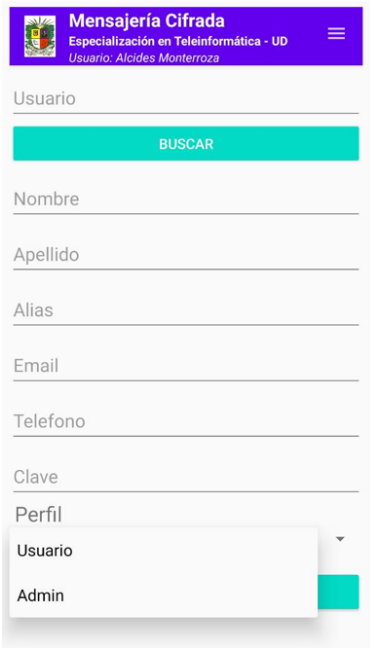


Figura 15. Opción del Menú Administrador.

- **Flujo de envío y recepción.**

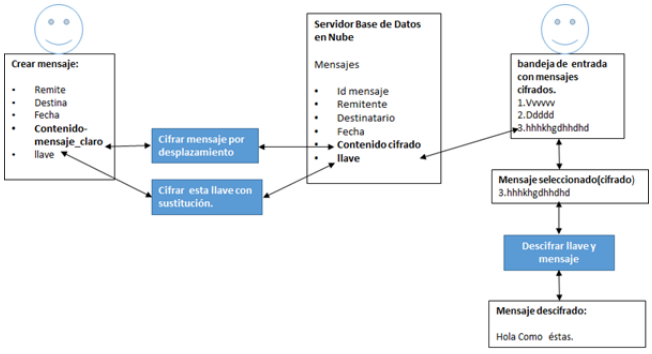


Figura 16. Flujo de Envío y Recepción.

1. Envío desde usuario 1 a usuario 2.
2. Almacenamiento del mensaje cifrado en la base de datos.
3. Verificar en usuario 2 mensaje de usuario 1.

- **API Rest – Web Service**

Se desarrollaron los servicios Rest para la aplicación, en Microsoft Visual Studio Community 2019 Versión 16.7.7, usando el lenguaje C#.

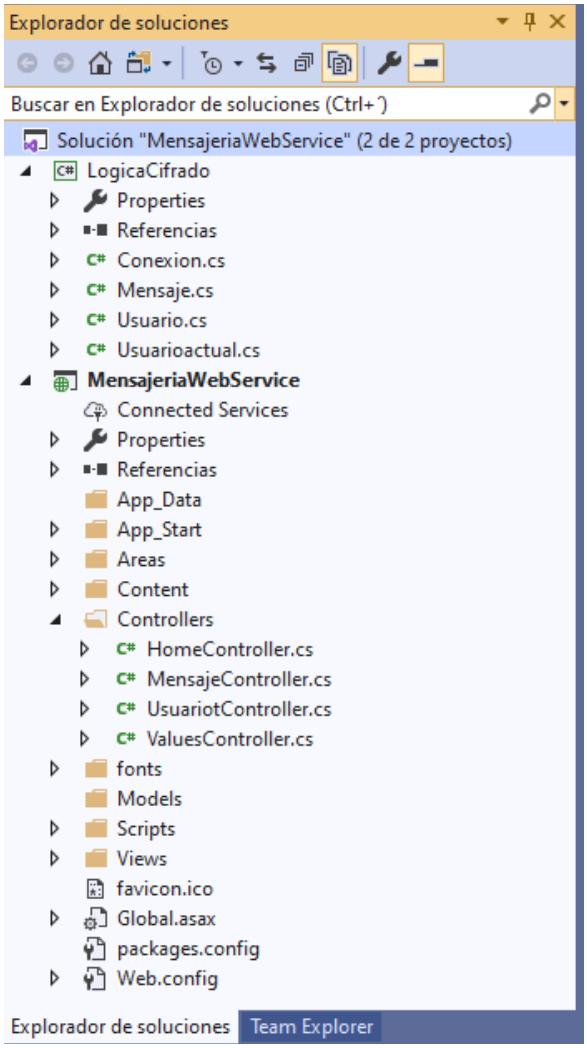


Figura 17. Proyecto API Rest en Visual Studio.

Se creó un proyecto de librerías llamado "LógicaCifrado", el cual contiene la lógica de conexión a la base de datos, y se agregó como referencia al proyecto llamado "MensajeríaWebService", el cual contiene las clases controlares y los métodos que finalmente se publican como servicios.

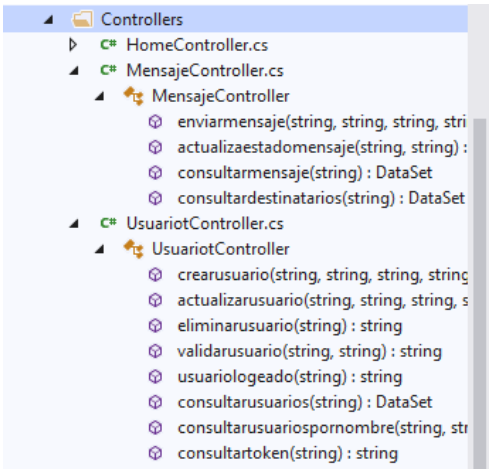


Figura 18. Métodos expuestos API Rest en Visual Studio.

Estos servicios se publicaron en la nube de Azure, en la siguiente *URL:* ***https://testud.azurewebsites.net/Help***

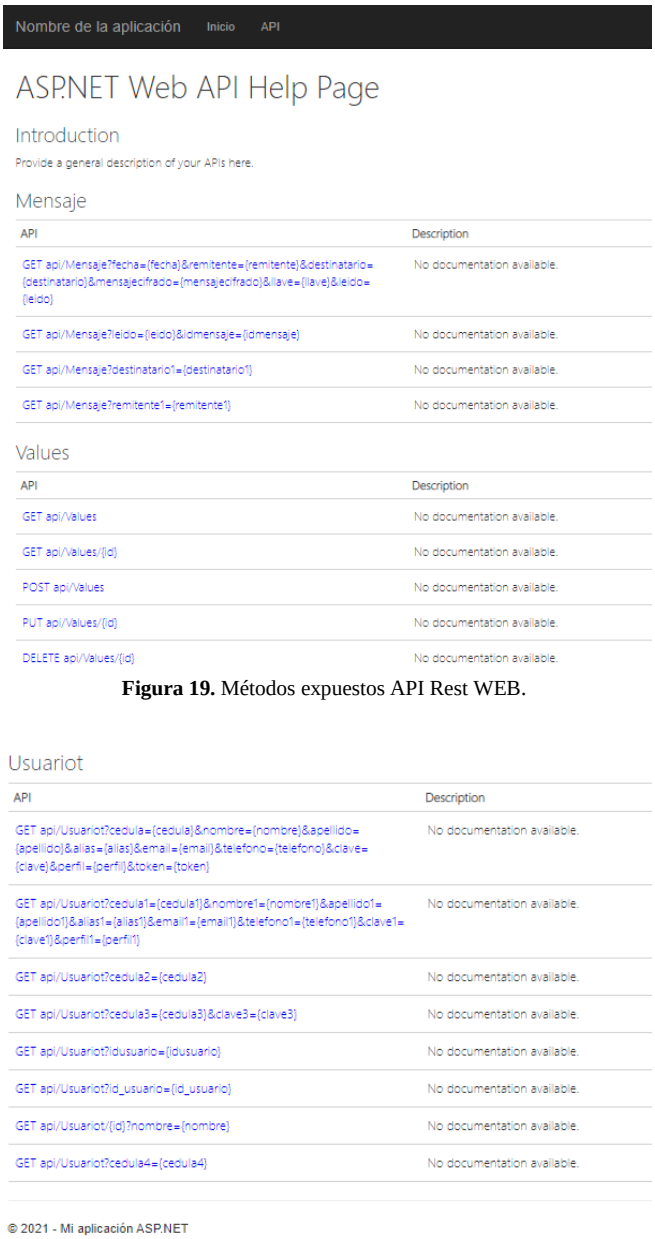


Figura 19. Métodos expuestos API Rest WEB.

Figura 20. Métodos expuestos API Rest WEB.

- **Base de datos**

Se creó una base de datos SQL en Azure, con dos tablas: mensaje y usuario.

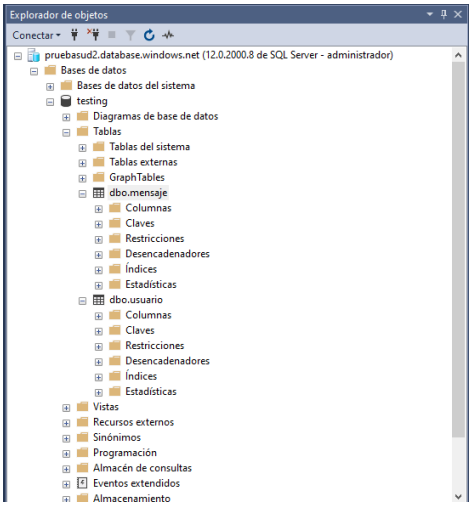


Figura 21. Tablas creadas en la DB motor MS-SQL.

En la siguiente imagen se observan los campos de cada tabla:

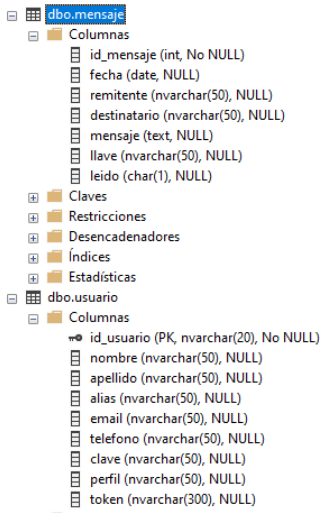
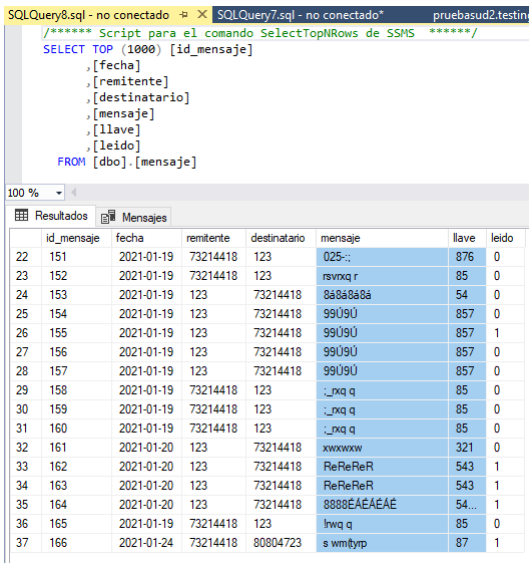


Figura 22. Campos existentes dentro de las tablas en MS-SQL.

Y finalmente de la siguiente forma queda almacenado el mensaje cifrado por desplazamiento en la base de datos y la llave cifrada por sustitución:



```
SQLQuery8.sql - no conectado    SQLQuery7.sql - no conectado*    pruebasud2.testin
/****** Script para el comando SelectTopNRows de SSMS  ******/
SELECT TOP (1000) [id_mensaje]
      ,[fecha]
      ,[remitente]
      ,[destinatario]
      ,[mensaje]
      ,[llave]
      ,[leido]
  FROM [dbo].[mensaje]
```

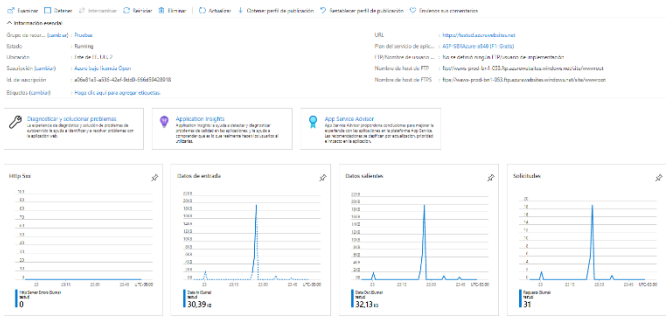
100 %

Resultados | Mensajes

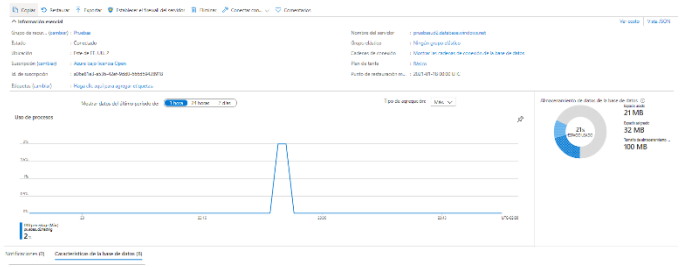
	id_mensaje	fecha	remitente	destinatario	mensaje	llave	leido
22	151	2021-01-19	73214418	123	025-:;	876	0
23	152	2021-01-19	73214418	123	rsvnxq r	85	0
24	153	2021-01-19	123	73214418	8á8á8á8á	54	0
25	154	2021-01-19	123	73214418	99Ú9Ú	857	0
26	155	2021-01-19	123	73214418	99Ú9Ú	857	1
27	156	2021-01-19	123	73214418	99Ú9Ú	857	0
28	157	2021-01-19	123	73214418	99Ú9Ú	857	0
29	158	2021-01-19	73214418	123	:_rxq q	85	0
30	159	2021-01-19	73214418	123	:_rxq q	85	0
31	160	2021-01-19	73214418	123	:_rxq q	85	0
32	161	2021-01-20	123	73214418	xwxwxw	321	0
33	162	2021-01-20	123	73214418	ReReReR	543	1
34	163	2021-01-20	123	73214418	ReReReR	543	1
35	164	2021-01-20	123	73214418	8888ÉÁÉÁÉÁÉ	54...	1
36	165	2021-01-19	73214418	123	lrwq q	85	0
37	166	2021-01-24	73214418	80804723	s wm{tyrp	87	1

Figura 23. Tablas creadas en la DB motor MS-SQL.

Revisión de API



4. Revisión en base de datos



Código Java desarrollado e implementado en el cliente, mediante el cual se realiza el cifrado y el descifrado del mensaje (por desplazamiento) y la llave (por sustitución).

```
package Clases;

import java.io.Serializable;

public class Cifradora{

    private  String mensajeclaro;
    private  String mensajecifrado;
    private  Integer valordesplazamiento;

    public void setMensajecifrado(String
mensajecifrado) {
        this.mensajecifrado = mensajecifrado;
    }

    public String getMensajecifrado() {  return
mensajecifrado;   }

    public String getMensajeclaro() {
        return mensajeclaro;
    }

    public void setMensajeclaro(String
mensajeclaro) {
        this.mensajeclaro = mensajeclaro;
    }

    public int getValordesplazamiento() {
        return valordesplazamiento;
    }
```

```
    public void
setValordesplazamiento(Integer
valordesplazamiento) {
        this.valordesplazamiento =
valordesplazamiento;
    }

    // Metodo para cifrar por
desplazamiento.
    public String
CifrarMensajedesplazamiento(){

        int longitud =
getMensajeclaro().length(); // se obtiene
la longitud del mensaje a cifrar
        char[] vectorcifrado = new
char[longitud]; // se define un vector de
caracteres donde se almacenará el mensaje
cifrado.
        char[] mimatriz = new char[] { 'A',
'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I',
'J', 'K', 'L',
                'M', 'N', 'Ñ', 'O', 'P',
'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X',
'Y', 'Z', 'a', 'b', 'c',
                'd', 'e', 'f', 'g', 'h',
'i', 'j', 'k', 'l', 'm', 'n', 'ñ', 'o',
'p', 'q', 'r', 's', 't',
                'u', 'v', 'w', 'x', 'y',
'z', ' ', ',', ';', ':', '.', '_', '-',
'*', '!', '"', '$', '%',
                '(', ')', '=', '?', '¡',
'¿', '0', '1', '2', '3', '4', '5', '6',
'7', '8', '9','Ú','Ó','Í',

'É','Á','ú','ó','í','é','á','/', (char)13
}; // Se define un vector que contiene la
base de caracteres que se usan en un
mensaje.
        int posicion = 0;
        int nuevaposicion = 0;
        char letra_obtenida;

        for (int i = 0; i < longitud; i++)
// se establece un ciclo desde cero hasta
el valor de la longirud del mensaje.
        {

          char []  letrica =
(getMensajeclaro().substring(i,i+1)).toChar
Array(); // se almacena cada letra del
mensaje en el vector "letrica"

            for (int b = 0; b < 94; b++) //
con este ciclo se recorre el vector que
contiene la base de caracteres
            {

                if (
mimatriz[b]==letrica[0]) // se pregunta
si la letra del mensaje almacenada en
"letrica", coincide con alguno de los
caracteres

//que se encuentran en el vector de
caracrteres base.
                {
                    posicion = b; break;  //
si se cumple la condición se obtiene la
posición.
                }
            }

            nuevaposicion = (posicion +
getValordesplazamiento()) % 94; // de
acruardo al valor de desplazamiento
suministrado por el usuario

// se obtiene la nueva posición, esta es la
formula del cifrado cesar.
            letra_obtenida =
mimatriz[nuevaposicion];// se guarda la letra
obtenida
            vectorcifrado[i] =
letra_obtenida; // y se va agregando al
vector cifrado.
        }

        String  texto_msj_crifrado = new
String(vectorcifrado); // se convierte el
vetor cifrado en un String.
        return texto_msj_crifrado; // se
retorna el texto cifrado.
    }

    // Método para descifrar por
desplazamiento, funciona igual que el
anterior, pero el contenido del vector base
debe estar invertido.

    public String
DescifrarMensajedesplazamiento(){
        char[] mimatriz = new char[] {
(char)13,
'/','á','é','í','ó','ú','Á','É','Í','Ó','Ú',
'9', '8', '7', '6', '5', '4', '3', '2', '1',
'0', '¿', '¡', '?', '=', ')', '(', '%', '$',
'"', '!', '*', '-', '_', '.', ':', ';', ',',
' ', 'z', 'y', 'x', 'w', 'v', 'u', 't', 's',
'r', 'q', 'p', 'o', 'ñ', 'n', 'm', 'l', 'k',
'j', 'i', 'h', 'g', 'f', 'e', 'd', 'c', 'b',
'a', 'Z', 'Y', 'X', 'W', 'V', 'U', 'T', 'S',
'R', 'Q', 'P', 'O', 'Ñ', 'N', 'M', 'L', 'K',
'J', 'I', 'H', 'G', 'F', 'E', 'D', 'C', 'B',
'A' };

        int longitud =
getMensajecifrado().length();
        char[] vectordescifrado = new
char[longitud];
```

```
        int posicion = 0;
        int nuevaposicion = 0;
        char [] letrica_cifrada;
        char letra_obtenida;
        for (int i = 0; i < longitud; i++)
        {
            letrica_cifrada =
getMensajecifrado().substring(i,
i+1).toCharArray();
            //int asciiCode = (int)
letrica_cifrada;

            for (int b = 0; b < 94; b++)
            {
                if (mimatriz[b] ==
letrica_cifrada[0])
                {
                    posicion = b; break;
                }
            }

            nuevaposicion = (posicion +
getValordesplazamiento()) % 94;
            letra_obtenida =
mimatriz[nuevaposicion];
            vectordescifrado[i] =
letra_obtenida;
        }


        String texto_msj_descifrado = new
String(vectordescifrado);

        return texto_msj_descifrado;
    }

    // Método para cifrar por sustitución
    public String
CifrarLlaveporSustitucion(){

        int longitud =
String.valueOf(getValordesplazamiento()).le
ngth();//se obtiene la longitud del mensaje
a cifrar.
        char[] letras = new
char[longitud];//vector para guardar el
mesaje cifrado.

        for (int i = 0; i < longitud; i++)
// se utiliza el ciclo para recorrer el
mensaje a cicfrar
        {
            char []  letrica =
String.valueOf(getValordesplazamiento()).su
bstring(i, i+1).toCharArray();// se obtiene
cada letra del mensaje a cifrar

// y se almacena en el vector "letrica"
            switch (letrica [0]) // según
sea la letra.
            {
                case 'A': letras[i] = 'z';
break; // se reempaza por la letra que
corresponda según la base de letras escogida y
se agrega alvector "letras"
                case 'B': letras[i] = 'y';
break;
                case 'C': letras[i] = 'x';
break;
                case 'D': letras[i] = 'w';
break;
                case 'E': letras[i] = 'v';
break;
                case 'F': letras[i] = 'u';
break;
                case 'G': letras[i] = 't';
break;
                case 'H': letras[i] = 's';
break;
                case 'I': letras[i] = 'r';
break;
                case 'J': letras[i] = 'q';
break;
                case 'K': letras[i] = 'p';
break;
                case 'L': letras[i] = 'o';
break;
                case 'M': letras[i] = 'ñ';
break;
                case 'N': letras[i] = 'n';
break;
                case 'Ñ': letras[i] = 'm';
break;
                case 'O': letras[i] = 'l';
break;
                case 'P': letras[i] = 'k';
break;
                case 'Q': letras[i] = 'j';
break;
                case 'R': letras[i] = 'i';
break;
                case 'S': letras[i] = 'h';
break;
                case 'T': letras[i] = 'g';
break;
                case 'U': letras[i] = 'f';
break;
                case 'V': letras[i] = 'e';
break;
                case 'W': letras[i] = 'd';
break;
                case 'X': letras[i] = 'c';
break;
                case 'Y': letras[i] = 'b';
break;
                case 'Z': letras[i] = 'a';
break;
                case 'a': letras[i] = 'Z';
break;
```

```
                case 'b': letras[i] = 'Y';
break;
                case 'c': letras[i] = 'X';
break;
                case 'd': letras[i] = 'W';
break;
                case 'e': letras[i] = 'V';
break;
                case 'f': letras[i] = 'U';
break;
                case 'g': letras[i] = 'T';
break;
                case 'h': letras[i] = 'S';
break;
                case 'i': letras[i] = 'R';
break;
                case 'j': letras[i] = 'Q';
break;
                case 'k': letras[i] = 'P';
break;
                case 'l': letras[i] = 'O';
break;
                case 'm': letras[i] = 'Ñ';
break;
                case 'n': letras[i] = 'N';
break;
                case 'ñ': letras[i] = 'M';
break;
                case 'o': letras[i] = 'L';
break;
                case 'p': letras[i] = 'K';
break;
                case 'q': letras[i] = 'J';
break;
                case 'r': letras[i] = 'I';
break;
                case 's': letras[i] = 'H';
break;
                case 't': letras[i] = 'G';
break;
                case 'u': letras[i] = 'F';
break;
                case 'v': letras[i] = 'E';
break;
                case 'w': letras[i] = 'D';
break;
                case 'x': letras[i] = 'C';
break;
                case 'y': letras[i] = 'B';
break;
                case 'z': letras[i] = 'A';
break;
                case ' ': letras[i] = '/';
break;
                case '/': letras[i] = ' ';
break;
                case ',': letras[i] = '¿';
break;
                case ';': letras[i] = '¡';
break;
                case ':': letras[i] = '?';
break;
                case '.': letras[i] = '=';
break;
                case '_': letras[i] = ')';
break;
                case '-': letras[i] = '(';
break;
                case '*': letras[i] = 'á';
break;
                case 'ó': letras[i] = '%';
break;
                case '!': letras[i] = '$';
break;
                case '"': letras[i] = 'í';
break;
                case 'í': letras[i] = '"';
break;
                case '$': letras[i] = '!';
break;
                case '%': letras[i] = 'ó';
break;
                case 'á': letras[i] = '*';
break;
                case '(': letras[i] = '-';
break;
                case ')': letras[i] = '_';
break;
                case '=': letras[i] = '.';
break;
                case '?': letras[i] = ':';
break;
                case '¡': letras[i] = ';';
break;
                case '¿': letras[i] = ',';
break;
                case '0': letras[i] = '9';
break;
                case '1': letras[i] = '8';
break;
                case '2': letras[i] = '7';
break;
                case '3': letras[i] = '6';
break;
                case '4': letras[i] = '5';
break;
                case '5': letras[i] = '4';
break;
                case '6': letras[i] = '3';
break;
                case '7': letras[i] = '2';
break;
                case '8': letras[i] = '1';
break;
                case '9': letras[i] = '0';
break;
                default: letras[i] = ' '; break;
            }
        }
        String texto_msj_crifrado = new
```

```
String(letras); // Se convierte el vector letras a string
        return texto_msj_crifrado; // se retorna el texto cifrado.
    }
}
```

X. CONCLUSIONES

Se implementó una aplicación móvil de mensajería cifrada, aplicando los métodos clásicos de cifrado: sustitución y desplazamiento. La aplicación sugiere una forma de cómo funcionan las aplicaciones de mensajería actuales, y permite comprender la importancia del papel que juega la criptografía en la confidencialidad de estas formas de comunicación.

Académicamente fue una experiencia enriquecedora donde se pusieron en práctica conceptos y temas que hicieron parte de la especialización en Teleinformática, tales como:

- Aplicación de metodología de diseño de software para el levantamiento de requerimientos, identificación de casos de uso, diagrama clases y diagrama de despliegue, usando UML.
- Desarrollo y publicación de Web Services en C# y Azure.
- Consumo de Web Services desde Java y Android.
- Conexión a Bases de Datos SQL Server en la nube de Azure.
- Desarrollo de App Móviles en Android Studio.
- Aplicación de cifrado simétrico a mensajes de texto.
- Evidencia de la aplicación de la criptografía en aplicaciones de mensajería.

Referencias

[1] M. J. L. Lapuente, «hipertexto.info,» [En línea]. Available: http://www.hipertexto.info/.

[2] Wikipedia, «Wikipedia.org,» [En línea]. Available: https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_m%C3%B3vi

[3] H. Rodriguez, «horaciorodriguez.webnode.com.co,» [En línea]. Available: https://horaciorodriguez.webnode.com.co/submodulos/mensajer -electronica/.

[4] kaspersky, «latam.kaspersky.com,» [En línea]. Available: https://latam.kaspersky.com/resource-center/definitions/encryption.

[5] Ionos, «Digital Guide,» 07 08 2019. [En línea]. Available: https://www.ionos.es/digitalguide/servidores/seguridad/todo-sobre-los-metodos-de-encriptado/.

[6] 12 Marzo 2017. [En línea]. Available: http://unidad-2-criptografia.blogspot.com/2017/03/24-criptosistemas-de-clave-publica.html.

[7] T. J. 2, «TIC Jrge 2,» [En línea]. Available: https://sites.google.com/site/ticgorge2/actividades/cifrado-de-la-informacion.