Diseño e implementación de un sistema seguro para la automatización de procesos centralizados y en campo, de un segmento de pymes

John Orlando Hurtado Restrepo, Pedro Alejandro Vacca Castro
Universidad de los Andes, Bogotá Colombia
Departamento de Ingeniería de Sistemas y Computación
Maestría en Seguridad de la Información – MESI
Noviembre de 2021

1 Introducción

Según (Controls et al., n.d.) "En la actualidad, las Pymes y MiPymes de Colombia son un aproximado de 2.540.953, lo que representa un 90% de las empresas del país, según los datos tomados del Ministerio de Trabajo…". De este total, hay aproximadamente 300.000 pymes que son clientes de un operador móvil, sobre las cuales se hizo un estudio de las que tuvieran una necesidad de una solución de fuerza de ventas con tres condiciones: automatizada, en la nube y segura; el estudio indicó que 25% (75.000) tienen esta necesidad.

Adicionalmente, se hizo un análisis de operatividad y seguridad dando como resultado lo siguiente:

- Las Pymes en estudio tienen fuerza de ventas y/o procesos logísticos en campo. Por ejemplo, ventas, inventarios y análisis de la competencia.
- La operación de la fuerza de ventas y procesos logísticos se hacen manualmente.
- Las pymes segmentadas para este caso no tienen ningún esquema de seguridad de la información.
- Ninguna tiene soluciones en la nube.

Las pymes con la necesidad identificada presentan una vulnerabilidad, que permite el robo de información, porque los empleados hacen los procesos de ventas, inventarios y análisis de la competencia de forma manual en hojas de cálculo desde un celular o un computador. En este contexto un empleado podría alterar la información por descuido o a propósito, igualmente un atacante podría acceder a la infraestructura (computador local), para robar dicha información.

En promedio cada pyme del grupo estudiado tiene 8 empleados; cada empleado procesa la información manualmente, tomando datos desde un celular con hojas de cálculo o enviándolas directo por mensajería hacia un coordinador de la empresa, que se encarga de recibir la información, clasificarla y recopilarla en Excel en un computador personal. Por ejemplo, la figura No. 1 muestra el agendamiento de los empleados para las visitas de la siguiente semana.

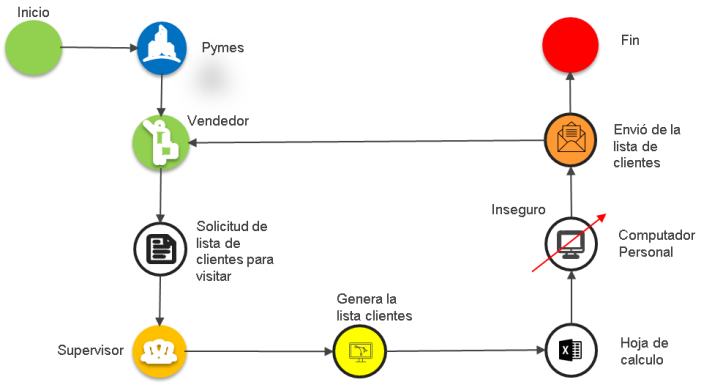


Figura 1. Situación actual de las pymes segmentadas.

Universidad de los Andes

2 Propuesta de solución

Este trabajo propone el diseño e implementación de un sistema seguro que permita automatizar los procesos que realizan los funcionarios en campo y procesos centralizados. El sistema debe proteger la información que intercambian los funcionarios en campo y la central, y adicionalmente debe ser diseñado de forma segura.

A continuación, se listan las funciones que debe cumplir la solución:

- Diseñar e implementar un sistema que permita garantizar la **integridad** de los datos.
- En el diseño se debe validar la **autenticidad** de las personas que pretendan hacer transacciones con el sistema.
- La **confidencialidad** de la información es otro factor que debe estar presente en el diseño de la solución propuesta.
- Adelantar un análisis de vulnerabilidades y plantear un plan para evitarlas y/o resolverlas.
- Garantizar los empleados de las Pymes puedan seguir usando el sistema, aún si se acaban el plan de datos.
- Garantizar a los empleados de las Pymes, a los que se les pierda o les roben el celular, no perder la información y seguir trabajando con la misma información al tener otro celular.

3 Diseño

Esta sección presenta el diseño de una aplicación que garantiza confidencialidad, integridad y disponibilidad en las comunicaciones que realiza un agente de campo con una oficina central por medio de celular.

3.1 Modelo de Amenazas

Las principales amenazas que deben ser manejadas por la aplicación son:

- Escaneo de puertos: desde celular se puede usar aplicaciones que escaneen los puertos, pero sólo podrán escanear los de whatsapp.
- Inyección SQL: desde whatsapp podrían intentar insertar código de SQL.
- Troyanos: podrían ejecutar un virus en el celular para ejecutar acciones malignas, como borrar información.
- Ransomware: se podrían robar la información del celular, pero la de whatsapp está cifrada.

3.2 Requerimientos Funcionales

- La aplicación debe permitir solo usuarios válidos, los usuarios no válidos se deberán bloquear de forma inmediata.
- La aplicación debe protegerse de las amenzas identificadas.
- Los usuarios pueden enviar o recibir archivos, localizaciones y palabras.
- El usuario debe saber que si no está autorizado será bloqueado en la primera solicitud que haga.
- El usuario debe utilizar una aplicación de mensajería (WhatsApp) que sea fácil de usar, rápida y que pueda realizar actualizaciones automáticamente en equipos móviles (celular) o en el computador.
- El usuario puede solicitar soporte en línea desde el mismo Chat.

3.3 Requerimientos No Funcionales

- La disponibilidad debe ser de al menos 99.4% anual, de acuerdo al RPO, RTO, tamaño de la transacción y la clasificación de WhatsIA para peticiones, quejas y reclamos (PQR´s): dentro de la clasificación de aplicaciones (Disponibilidad continua, Misión crítica, Importante para la empresa y No tan importantes), WhatsIA se clasifica como Importante para la empresa, ya que un usuario puede enviar una solicitud y si la plataforma no responde, puede enviarlo de nuevo más tarde o enviarlo por correo electrónico, los

mantenimientos se harán el ultimo de cada mes, en el horario de 1 y 5 de la mañana. A continuación, se calculan los tiempos para el servicio 7/24:
Convenciones:

- ✓ MTTR: Tiempo medio de recuperación de fallas (Mean Time To Recovery).
- ✓ MTBF: Tiempo medio entre fallas (Mean Time Between Failures).
- ✓ RTO: Tiempo objetivo de recuperación.
- ✓ RPO: Punto objetivo de recuperación.
- ✓ RTS: Tiempo de recuperación del servicio.
- ✓ STS: Tiempo de solución de los servicios.

Máximo número y tiempo de recuperación de Fallas Menores**:**

$$Horas = 24$$
$$Dias = 30$$
$$MTTR\ (Minutos) = 60 = 1\ (hora)$$
$$Máximo\ número\ de\ fallas\ en\ los\ 30\ dias = 4$$

$$MTBF\ (Horas) = \frac{24(Horas) * 30(Dias)}{4\ (Fallas\ al\ mes)} = 180(Horas)$$
$$Disponibilidad = \frac{MTBF}{MTBF\ + MTTR} = 99{,}4475\ \%$$

(Sjaak Laan, n.d.)

RTO y RPO para Fallas Intermedias**:**

$$RTO = \ 12\ Horas$$
$$RPO = 6\ Horas$$

RTO y RPO para Fallas Mayores**:**

$$RTO = 15\ Horas$$
$$RPO = 6\ Horas$$
$$RTS = 24\ Horas$$
$$STS = 24\ Horas$$
$$Tiempo\ para\ atender\ y\ escalar\ servicio: 90\ minutos$$
$$Tiempo\ de\ conservación\ de\ datos: 6\ meses$$
$$Porcentaje\ de\ backup\ exitoso: 90\%$$

(EMC Education Services, 2012)

- El sistema debe ser transaccional y los tiempos de respuesta dependen, por un lado, de aplicación de mensajería del celular y por el otro el tiempo de respuesta del sistema, los tiempos que no incluyen archivos son menores a 5 segundo y los que incluyen archivos son menores a 10 segundos. No se admiten archivos mayores a 250 KB.
- La información que envían los empleados debe enviarse en forma segura y almacenarse en un data center con los principios básicos de seguridad.
- La información almacenada de cada Pyme debe ser consultada a través de un sistema de mensajería seguro.

3.4 Requerimientos Técnicos

- Los usuarios deben usar celulares que permitan el uso de WhatsApp en principio.
- Los usuarios necesitan tener datos o WIFI o WhatsApp gratis.
- Se requiere un motor de base de datos MySQL.
- El sistema operativo que soporta la solución es Windows server 2016 en adelante.
- El ancho de utilizado debe ser 20 Mbps en principio.

Universidad de los Andes

3.5 Principios de Diseño Seguro

Se quiere brindar una herramienta automatizada y segura (que esté libre de vulnerabilidades como SQL Injection, Cross-site Scripting (XSS), Phishing y Spam, Insecure Object References, Path traversal Cross-site Request Forgery, Security misconfiguration, Broken Authentication, XML external entities), que le permita a cualquiera de las Pymes identificadas hacer su trabajo en campo desde cada celular sin instalar aplicaciones, sin importar que se acaben los datos de su plan y sin costo adicional a este.

Para proteger la aplicación de las amenazas identificadas se seleccionó un esquema de "limpieza" de entradas basado en listas blancas. Adicionalmente, se redujo la superficie de ataque al mínimo. A continuación, se explican estos principios de diseño.

Limpieza de entradas:

La aplicación solo permite el paso de patrones identificados como benignos y los demás son rechazados.

- Al usuario sólo se le permite escribir palabras válidas que previamente han sido matriculadas en el sistema y que hayan sido asociadas al usuario que pretende usarlas.
- El usuario tendrá un menú de palabras válidas que tienen una funcionalidad. por ejemplo, un usuario puede escribir la palabra "CVentas", que le permite consultar sólo las ventas de él, esa misma palabra escrita por un supervisor permite la consulta de las ventas de todos los empleados.
- A cada palabra clave se le asocia una funcionalidad y un rol de usuario. Cada palabra puede corresponder a un proceso de fuerza de ventas; por ejemplo, el usuario envía la palabra venta y el sistema se en encarga de devolverle el formulario que debe diligenciar para este proceso.
- Transformar las funcionalidades hechas con lenguajes de programación como SQL, JavaScript, PHP entre otros, en palabras claves usadas desde la aplicación de mensajería para evitar ataques como SQL Injection, Cross-site Scripting (XSS), Phishing y Spam, Insecure Object References, Path traversal Cross-site Request Forgery, Security misconfiguration, Broken Authentication, XML external entities.

Reducción de la superficie de ataque:

- Evitar que los usuarios puedan hacer análisis de vulnerabilidades y su explotación, haciendo que el servidor central no exponga servicios adicionales al de la aplicación de mensajería.
- No se tendrá expuesto ningún servicio (puerto), internamente solo tendrá expuesto el servicio de escritorio remoto a través de un puerto valido. Esto evitaría el escaneo de puertos y por ende la exploración de muchas vulnerabilidades y su explotación.
- Solo se autorizarán las Ip's de los administradores que se conecten por escritorio remoto.
- Cada proceso se debe comunicar con el módulo de seguridad de la base de datos a través de una dirección Ip válida y un hash.
- Cada funcionalidad tiene un proceso manejado por una pieza de software diferente, dejándole al sistema operativo el manejo de asignación de procesamiento de acuerdo con el número de núcleos, para garantizar el paralelismo.

La siguiente tabla analiza las ventajas ofrecidas por el diseño contra las amenazas identificadas previamente.

Tipo	Subtipo	Descripción
Network Scanning		Como el sistema no expone ningún servicio o puerto y tampoco tiene una ip pública, un atacante no podrá hacer peticiones porque no hay donde hacerlas.
Abuso	Bruteforce Attacks	Este tipo de ataques, que atentan contra el control de acceso de un sistema por medio de múltiples intentos de autenticación a uno o varios usuarios a través del uso de un número variado de contraseñas, no aplica en esta solución ya que no se solicita al empleado ni usuario ni

Tipo	Subtipo	Descripción
		contraseña, el usuario es tomado automáticamente y no hay lugar a cambiarlo.
	SQL Injection	Si un atacante quisiera hacer un SQL Injection, no podría ya que no hay ninguna puerta abierta hacia el motor de base de datos y no se puede escribir código en WhatsApp diferente a comandos preestablecidos.
Malware	Troyano	Este software que afecta la confidencialidad del equipo ejecutando acciones y robando información del sistema, no podrá cumplir su cometido ya que cada mensaje generado en WhatsApp está cifrado extremo a extremo con una llave diferente y con un algoritmo forward, que no permite hacer análisis hacia atrás; así está cifrada la base de datos local, que es lo único que un atacante podría robarse, pero no descifrarla.
	Rasomware	Lo único que podrían atacar es un celular, cifrando el almacenamiento, en tal caso, nunca afectará el sistema central y la solución es individual y tiene que ver con todo el teléfono, no específicamente con WhatsApp.

Tabla 1. Modelo de amenazas

Adicionalmente, el nivel de riesgo de la protección de la información cambia con el diseño propuesto, ya que la información no estará almacenada en los celulares de los agentes en campo, en vez de esto la información estará almacenada en un centro de cómputo Tier IV con todos los principios básicos de seguridad.

3.6 Arquitectura y componentes del sistema

La aplicación funciona en un esquema cliente/servidor donde la aplicación de mensajería del celular es el cliente y hay un sistema central que se encarga de asumir el rol de servidor.

Cliente:

La aplicación de mensajería debe implementar autenticación de usuario y cifrado punto a punto. WhatsApp cumple con estos requerimientos; puede estar protegido por clave, huella o reconocimiento facial (**Primer punto de seguridad**: si se roban el teléfono no se puede hacer transacciones).

Servidor:

Existen tres sistemas en la arquitectura, el sistema central, el sistema de procesos y el sistema de diseño:

- **Sistema central,** se compone de cuatro subsistemas asincrónicos que funcionan de forma independiente, estos son:
  - El subsistema de seguridad de base de datos (**SDB**), el cual recibe, de procesos autorizados, cualquier solicitud que necesite interactuar con la base de datos (**Segundo punto de seguridad**: IdProceso e IP válidos).
  - El subsistema de lectura (**SL**), recibe los mensajes, analiza que el usuario (número de WhatsApp tomado automáticamente) sea válido (**Tercer punto de seguridad:** no se permiten usuarios no válidos y no se requiere autenticación), analiza que el mensaje sea válido (**Cuarto punto de seguridad:** sólo admite mensajes matriculados) y lo entrega a un subproceso **SDB**.
  - El subsistema de análisis de mensajes (**SAM**) toma los mensajes por orden de llegada, determina si el usuario que lo envió tiene permisos para ejecutarlo (**Quinto punto de seguridad**), igualmente analiza

a qué proceso corresponde según el usuario que lo envía, hecho esto, el mensaje es enviado al sistema de procesos.

- Subsistema de escritura (**SE**), es un proceso que lee los mensajes pendientes por enviar a los usuarios, la lectura es tomada en orden de llegada y es enviada al usuario correspondiente.

- **Sistema de procesos,** es un sistema compuesto por múltiples subsistemas, cada uno tiene una tarea específica, los procesos son: procesos de documentos, de formularios, de georreferenciación, de machine learning, de Contac center, de consultas y de menús.
  Cada subsistema está matriculado en la **DB** con una identificación y una dirección Ip, esto con el fin de validar desde el subproceso **SDB**, que el proceso esté permitido (**Sexto punto de seguridad**).
  La información resultante es guardada en la **DB** para que pueda ser tomada por el sistema de escritura (**SE**).
- **Sistema de diseño,** con este sistema se parametrizan las entradas, salidas y funcionalidad, bajo un nombre al que llamamos aplicación, por ejemplo, la aplicación de ventas tiene como entrada referencia y valor, esos datos deben ser insertados en la DB. Dentro de cada aplicación se parametriza el mensaje con el que se puede acceder y los roles de usuarios que pueden ejecutar la aplicación.

Con este diseño se garantiza, primero, la seguridad de la información, previniendo las vulnerabilidades y su explotación; segundo, el procesamiento en paralelo; tercero, independencia de todos y cada uno de los subprocesos; cuarto, independencia en hardware, ya que cada subsistema podría ejecutarse en servidores independientes; y quinto, independencia de lenguajes de programación, ya que cada subproceso puede desarrollarse en lenguajes diferentes. En las figuras 2, 3 y 4 se muestra esta arquitectura.

Entrada de información desde WhatsApp al sistema.

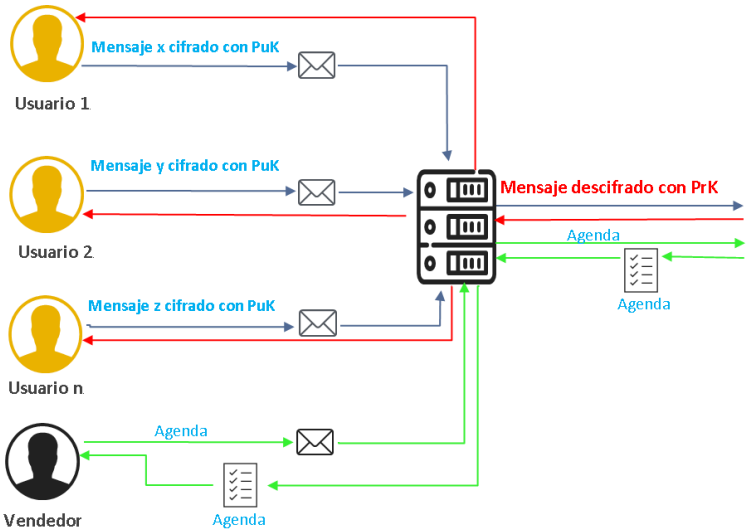


Figura 2. Arquitectura del sistema externo (Ramírez, n.d.)

Sistemas de lectura, procesos y escritura de los mensajes que se reciben de whatsapp.

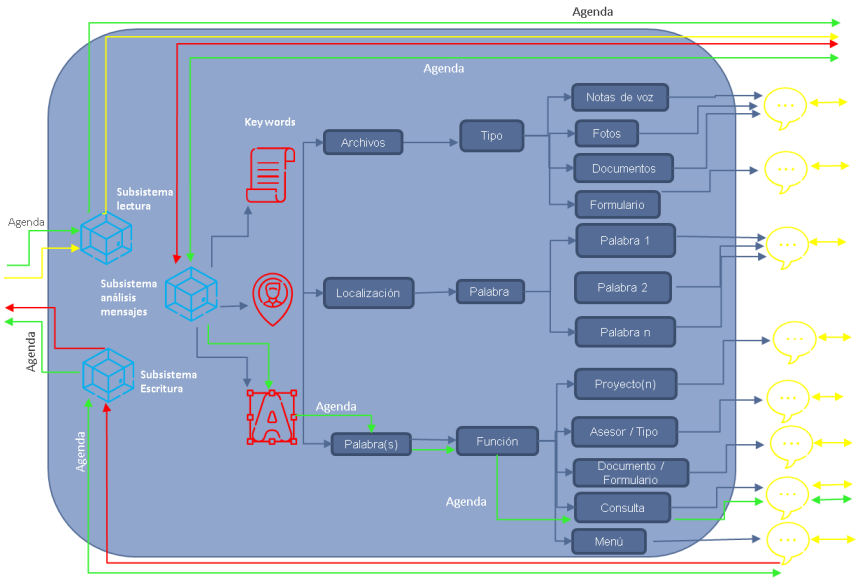


Figura 3. Arquitectura del sistema central

Sistema de procesos que se encarga de interactuar con la base de datos.

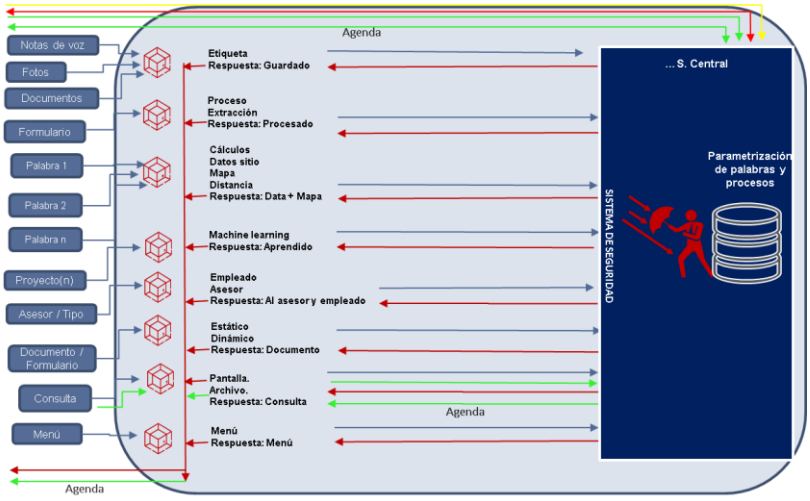


Figura 4. Arquitectura del sistema de procesos

3.7 Diagrama de infraestructura

La empresa WHATSIA montará esta solución sobre un Data Center TIER IV donde se desplegará el sistema que contará con dos servidores principales: Servidor del sistema central y servidor del sistema de procesos, sistema de seguridad y la base de datos (MySQL), los cuales poseen especificaciones estándar del centro de cómputo. El ecosistema se montará como IaaS (Infraestructura como un servicio) que será desplegado sobre un conjunto de soluciones de hardware, software, herramientas de gestión, procesos y procedimientos, también se manejará un Storage para la ejecución de backup's de las máquinas virtuales.

En la figura 5 se observa el diagrama de la infraestructura.

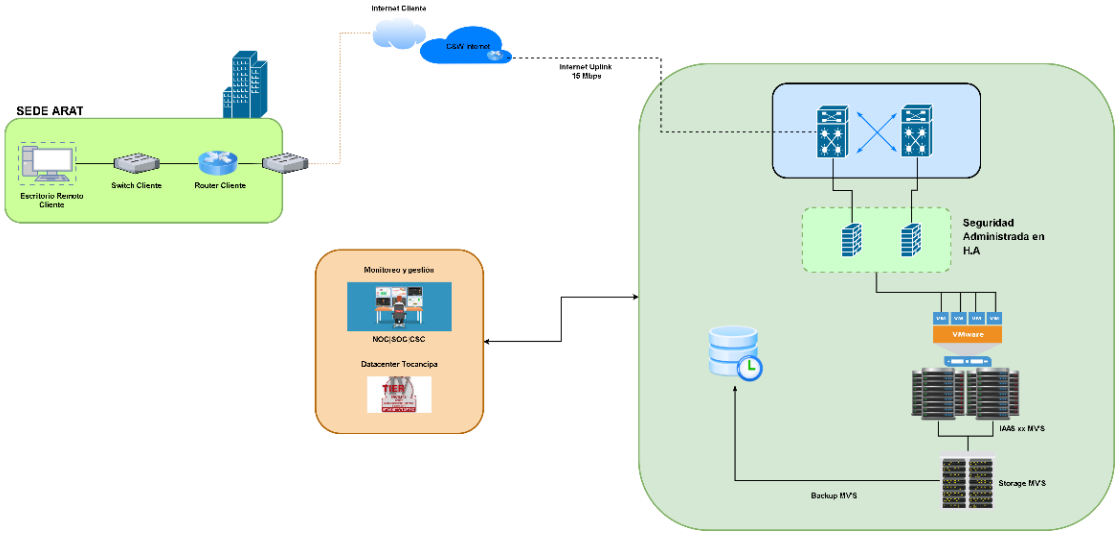


Figura 5. Diagrama de infraestructura.

4 Implementación

4.1 Lenguajes y herramientas usados para la programación.

Para la programación del núcleo, es decir, el sistema central y los subsistemas, se usaron los lenguajes de programación C y C++ en el sistema operativo Windows 10; para procesos que no son del núcleo se usó el lenguaje Python y el motor de base de datos usado fue MySQL corriendo en un Windows Server 2016.

Las herramientas usadas fueron C++ Builder 10.3 y Visual Studio Code.

4.2 Integración de los sistemas de seguridad al sistema general

Para realizar la descripción de todos y cada uno de los subsistemas, se tuvo en cuenta el numeral *3.6 Arquitectura y componentes del sistema*; lo cual permitió realizar la descripción de la implementación de los puntos de seguridad.

4.2.1 Implementación del segundo y sexto punto de seguridad

Para esta implementación, se tomó como referencia el subsistema de seguridad de base de datos (**SDB**); primero se creó un usuario que es el único autorizado a interactuar con la base de datos con los permisos select, insert, update y delete; luego se creó una tabla que contiene los campos proceso, direccionip y clave.

El sincronismo se maneja con estados, un estado es un valor manejado en la tabla de mensajes, que determina si un mensaje se debe enviar a un usuario o se debe procesar internamente.

La lógica que se implementó en lenguaje C fue un listener por un puerto del sistema de procesos, en el que se reciben las consultas, se verifica que el proceso sea válido y si lo es, se ejecuta la consulta recibida afectando los registros correspondientes. Hay dos tipos de respuestas: 1. Internas, estas devuelven un mensaje inmediato al proceso que la solicitó y 2. Externas que escriben una respuesta en la tabla mensajes para que posteriormente el sistema de escritura lo tome y lo envíe al usuario. A continuación, se presentan tres ejemplos:

Ejemplo de respuesta interna: un proceso solicita la validación de un usuario, la respuesta es inmediata.

Ejemplo 1 de respuesta externa: si se reciben los datos para registrar una venta, el sistema valida con la tabla de parámetros si los datos son válidos, si es así, se inserta un registro con los datos recibidos y se escribe en la tabla de mensajes la respuesta exitosa.

Ejemplo 2 de respuesta externa: si se reciben datos para una consulta, se verifican en la tabla de parámetros los datos recibidos, si están bien, se ejecuta una consulta y el resultado se deja en la tabla de mensajes con un estado 0.

En cualquiera de los ejemplos de respuesta externa, si los datos no son válidos, se escriben en la tabla de mensajes la respuesta negativa.

4.2.2 Implementación del tercer y cuarto punto de seguridad

Se basa en el diseño del subsistema de lectura (**SL**) y consiste en tomar el mensaje entrante de un usuario y hacer el proceso en dos pasos, primero, lanza una consulta para validar el usuario (número de WhatsApp tomado automáticamente) al sistema SDB, así se implementa el tercer punto de seguridad descrito en la arquitectura (no se permiten usuarios no válidos y no se requiere autenticación); el segundo paso es lanzar la consulta para garantizar que el mensaje sea válido, cumpliendo así con el cuarto punto de seguridad de la arquitectura (sólo se admiten mensajes matriculados).

4.2.3 Implementación del quinto punto de seguridad

La base de esta implementación es el subsistema de análisis de mensajes (**SAM**), que toma los mensajes por orden de llegada y determina si el usuario que lo envió tiene permisos para ejecutar ese mensaje específico; esto cumple la implementación del quito punto de seguridad.

4.2.4 Implementación del asincronismo en la operación

La forma en que operan todos los subsistemas es asíncrona, esto se logró a través de la inclusión del campo *estado* en todas las tablas de la base de datos que intervienen en el proceso, además se creó un protocolo de comunicaciones estilo token, en la que cada proceso verifica sus estados pendientes por atender, por ejemplo, el subsistema de escritura (**SE**) se comunica permanentemente con el sistema (**SDB**) preguntándole por estados de mensajes pendientes por enviar, en orden de llegada, si hay alguno, toma el mensaje y lo envía al usuario correspondiente.
De esta forma se garantiza que la entrada de mensajes sea totalmente independiente de la salida y sus tiempos no deben estar sincronizados, de tal forma que puede haber muchos mensajes de entrada al tiempo y sus respuestas se harán por sistema de colas.

4.2.5 Implementación de la funcionalidad y su seguridad

Para facilidad de los usuarios se implementaron las funcionalidades de ventas, inventarios y análisis de la competencia en una sola aplicación; esto se logró creando un formulario con el sistema de diseño llamado el Núcleo; en este sistema se pueden crear clientes, en cada cliente se pueden crear proyectos y en cada proyecto se pueden crear aplicaciones, por ejemplo, la aplicación que se implementó corresponde al cliente al proyecto Contactabilidad y se llama Ventas_Inventario_Competencia (ver Figura 6).

El formulario está compuesto por:

- Un campo oculto donde se guardará el SHA-3, que está compuesto por un secreto que involucra el número del celular, un número aleatorio con rangos diferentes en cada caso y algunos dígitos de precisión aleatorios de los segundos tipo Unix (desde 01/01/1970).
- Un encabezado, en el que se muestran los datos del usuario que solicitó dicho formulario, es decir, es personalizado.
- Una lista de puntos de venta donde el usuario seleccionará el punto en que se hace la operación.
- Una lista de selección múltiple y editable en la que se muestran todas las referencias de los productos de la empresa que se vendan en ese punto y en cada referencia se puede entrar las cantidades vendidas.
- Una lista de selección múltiple para reportar los inventarios por referencias.

- Una lista de selección múltiple para reportar las ventas globales de cada fabricante de la competencia.
- Un campo abierto para que se escriban observaciones.

A continuación, se muestra el diseño real hecho con el módulo de parametrización; en este caso se creó un formulario que contiene las ventas, los inventarios y las ventas de la competencia

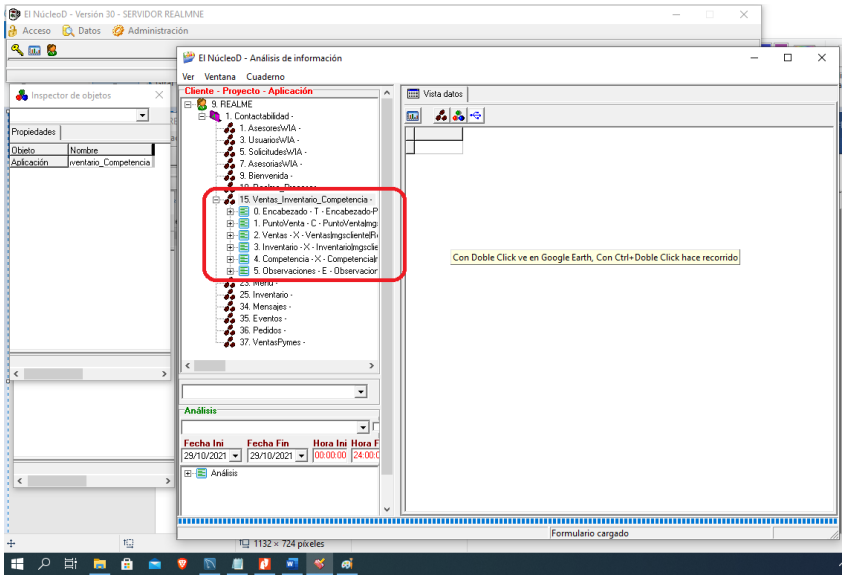


Figura 6. Implementación de Ventas, Inventarios y Análisis de la Competencia.

4.2.6 Proceso

El ciclo de vida del proceso es el siguiente:

1. El usuario escribe en el chat la palabra **FVentas** (ver Figura 7), para solicitar el formulario de Ventas_Inventarios_Competencia.

   El sistema crea un formulario al instante y lo envía, este formulario es personalizado con el nombre, celular del solicitante y el SHA-3, cuyo contenido incluye un secreto que involucra el número del celular, un número aleatorio con rangos diferentes en cada caso y algunos dígitos de precisión, que corresponde a los segundos tipo Unix del momento en el que se genera el formulario para ser enviado, finalmente se codifica en base 64. Se incluyó el número aleatorio para que el hash no sea predecible. Cada vez que el usuario solicita un formulario o el formulario cambia por que cambian sus referencias o puntos de venta, se genera otro SHA-3.

   En la tabla de usuarios, se actualiza el campo de tiempo (segundos) que hizo parte del SHA-3 para que, al recibir el formulario diligenciado, el sistema genere el SHA-3 y lo compare con el que llega para comprobar si es auténtico; con esto se garantiza que, si un usuario diferente al que solicitó el formulario, lo envía, éste no se procesará (ver Figura 8).

2. Al recibir el formulario, el usuario lo abrirá y llenará con las ventas, inventarios y ventas de la competencia (ver Figura 8).

3. El usuario enviará el formulario diligenciado (ver Figura 7) y el sistema responderá “Solicitud procesada”. Esta respuesta implica validar internamente todos los datos y validar que ese formulario pertenece a ese usuario validando el sha-3 que es único por usuario y formulario

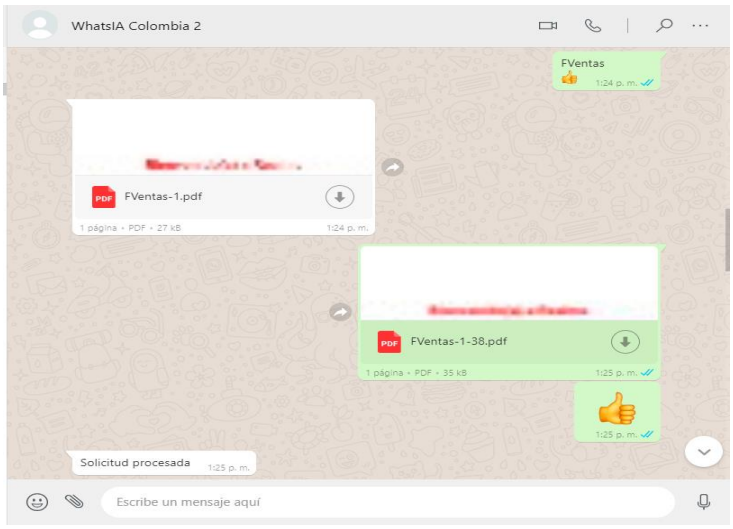
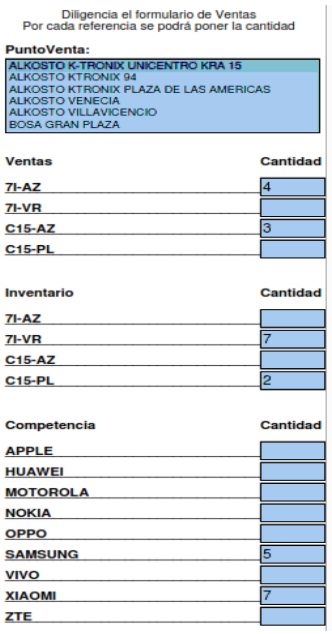


Figura 7. Solicitud y envío real del formulario.



Diligencia el formulario de Ventas
Por cada referencia se podrá poner la cantidad

PuntoVenta:

ALKOSTO K-TRONIX UNICENTRO KRA 15
ALKOSTO KTRONIX 94
ALKOSTO KTRONIX PLAZA DE LAS AMERICAS
ALKOSTO VENECIA
ALKOSTO VILLAVICENCIO
BOSA GRAN PLAZA

Ventas	Cantidad
7I-AZ	4
7I-VR	
C15-AZ	3
C15-PL	

Inventario	Cantidad
7I-AZ	
7I-VR	7
C15-AZ	
C15-PL	2

Competencia	Cantidad
APPLE	
HUAWEI	
MOTOROLA	
NOKIA	
OPPO	
SAMSUNG	5
VIVO	
XIAOMI	7
ZTE	

Figura 8. Formulario real diligenciado.

5 Resultados

Se hicieron operaciones reales con 194 usuarios ubicados geográficamente en diferentes partes de Colombia, desde el 2 de septiembre hasta el 12 de octubre del 2021, generando un total de 11.418 registros.

Para efectos prácticos se muestra parte de los resultados del día 28 de septiembre del 2021 protegiendo las columnas de datos personales como el celular y los nombres; en este resultado se evidencia que las referencias del formulario son las mismas, al igual que los nombres de los fabricantes de la competencia y los nombres de los puntos de venta que están en el formulario de la figura No. 8.

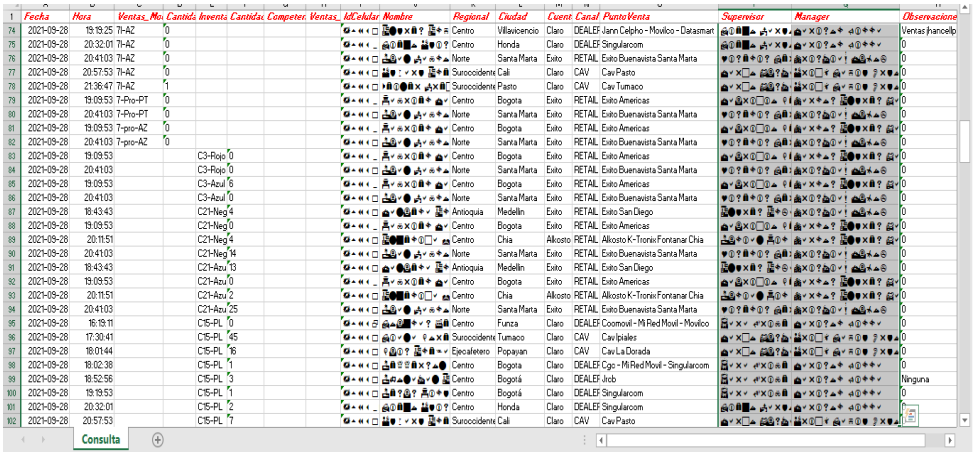


Figura 9. Resultados reales.

Algunos usuarios hicieron pruebas desde celulares que no estaban registrados y el sistema los rechazó.

Otros usuarios trataron de enviar formularios de algún compañero y se rechazó (ver Figura 10).

Algunos enviaban palabras no autorizadas y el sistema los rechazó (ver Figura 11).

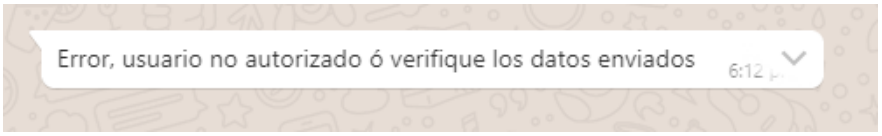


Figura 10. Rechazo de usuario que envió formulario de compañero.

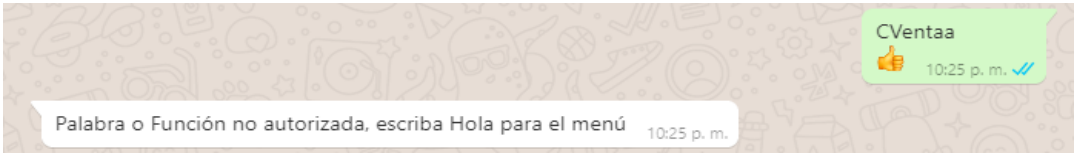


Figura 11. Rechazo de una palabra no autorizada.

Se intentó implementar ataques, pero las herramientas de análisis de vulnerabilidades por medio de puertos, como Owasp, Nmap y otras, requieren una dirección IP expuesta y la aplicación implementada no presenta puertos abiertos. Por otro lado, antes de salir a producción el próximo año, se hará un segundo análisis más especializado.

Se preparó un robot de envío de mensajes cada segundo al celular de la solución y el sistema lo bloqueará en el primer intento ya que en la implementación bloquea celulares que no están registrados.

Se aclara que el alcance de la seguridad no llega hasta la protección del celular de cada usuario, por lo tanto, si a algún usuario le toman el celular y desde ahí hacen transacciones válidas, el sistema lo tomará como verdadero, aunque el usuario puede avisar inmediatamente al operador móvil para que el celular sea bloqueado.

6 Conclusiones y Aportes

6.1 En cuanto al diseño

El diseño hecho e implementado para el problema planteado, soluciona tanto la necesidad que tiene el operador con las Pymes, como la seguridad de la información para todas y cada una de las empresas. La evidencia es que fue seleccionado entre más de 20 empresas que propusieron soluciones diferentes.

Se implementó seguridad en diferentes puntos de la arquitectura, impidiendo el análisis de vulnerabilidades por medio de herramientas que automatizan el escaneo de puertos.

Se adicionó un factor de autenticación de la información que envía el usuario a través de los formularios con la primitiva SHA-3 que garantiza, que un usuario no puede enviar información a nombre de otro.

La elección de una herramienta que interactúe con el sistema como WhatsApp tiene varias ventajas:

- Protocolo de cifrado extremo a extremo garantizado por una de las empresas más grandes y seguras del mundo, como lo es Meta.
- Valores cero: costos de capacitación, costos de instalación, actualizaciones automáticas, tiempos de desplazamiento para instalaciones.
- Las pymes pueden automatizar sus procesos en campo sin costo adicional, ya que está incluido en el plan; lo que le interesa al operador es no dejar ir a sus clientes.
- Los usuarios que se queden sin datos, pueden seguir operando sin problema, ya que el plan que el operador incluye whatsapp ilimitado, así se acaben los datos

6.2 A nivel de requerimientos

En la aplicación, a nivel de seguridad se garantiza la autenticidad porque se valida que el número de WhatsApp esté registrado en el sistema y no puede haber dos números iguales en una red celular.

Otro punto de seguridad fuerte en la implementación es cuando se asocian palabras claves con una funcionalidad y un rol de usuario, donde cada palabra puede corresponder a un proceso de fuerza de ventas evitando ataques de programación como SQL Injection, Cross-site Scripting (XSS), Phishing y Spam, Insecure Object References, Path traversal Cross-site Request Forgery, Security misconfiguration, Broken Authentication, XML external entities.

El último punto de seguridad es evitar que los usuarios hagan análisis de vulnerabilidades con herramientas como OWASP, Metasploit, Nmap y Kismet, porque la única aplicación expuesta es WhatsApp y la implementación no se expone en ningún servicio.

6.3 En la implementación

Se garantiza en la implementación, la agilidad de operación y la facilidad de interactuar con el sistema, a la vez que permite al usuario la tranquilidad que ninguna otra persona puede hacer transacciones a su nombre, siempre y cuando el usuario habilite las operaciones de protección en el cliente.

6.4 Aportes

Consideramos que el aporte más importante, es el diseño que es modular, asíncrono y paralelo, que permite desarrollar mucha más funcionalidad en diferentes lenguajes de programación, conservando la seguridad, ya que cualquier pieza de software que quiera interactuar con el sistema debe estar plenamente identificada y autorizada.

La seguridad implementada, es una nueva propuesta de seguridad que puede servir en sistemas cuya herramienta de trabajo sean chats universales como Signal, Telegram y WhatsApp.

6.5 Mejoras futuras

Debido a que el diseño es modular y cada pieza puede ser desarrollada en diferentes lenguajes, otros grupos de investigación podrían tomar este trabajo e implementar más funcionalidad y proponer nuevos esquemas de seguridad que protejan aún más a las pymes.

Universidad de los Andes

7 Referencias

Controls, C., Practices, G., Enterprise, M., López, G., Camilo, J., & Mesa, R. (n.d.). Guía de Controles y Buenas Prácticas de Ciberseguridad para MiPymes.

EMC Education Services. (2012). Information Storage and Management: Storing, Managing, and Protecting Digital Information in Classic, Virtualized, and Cloud Environments, 2nd Edition. http://aad.tpu.ru/practice/EMC/Information%20Storage%20and%20Management-v.2.pdf

Eset. (2019). Eset Security Report 2018: el estado de la seguridad de la información en las empresas de la región. https://www.welivesecurity.com/la-es/2018/06/19/eset-security-report-2018-el-estado-de-la-seguridad-de-la-informacion-en-las-empresas-de-la-region/

Eset. (2020). SECURITY REPORT LATINOAMÉRICA 2020. https://www.welivesecurity.com/wp-content/uploads/2020/08/ESET-Security-Report-LATAM_2020.pdf

Ingeniería de sistemas y software - Requisitos y evaluación de calidad de sistemas y software (SQuaRE) - Modelos de calidad de sistemas y software. (n.d.). Retrieved October 5, 2021, from https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en

Juan Manuel Wilches Durán, & Germán Darío Arias Pimienta. (n.d.). Normograma del Ministerio de Tecnologías de la Información y las Comunicaciones [RESOLUCION_CRC_4930_2016]. Retrieved September 14, 2021, from https://normograma.mintic.gov.co/mintic/docs/resolucion_crc_4930_2016.htm

Mejía Jiménez Genesis Yamilet, & Romero Arévalo Shirley Lisbeth. (2019). Análisis de vulnerabilidad en un sistema de gestión de seguridad de información en una pyme del sector comercial. http://repositorio.ug.edu.ec/bitstream/redug/44386/1/TESIS%20MEJIA%20Y%20ROMERO%202019.pdf

Ramírez, I. (n.d.). WhatsApp vs Telegram vs Signal, comparativa: ¿cuál es la app de mensajería más segura? Retrieved August 27, 2021, from https://www.xatakandroid.com/seguridad/whatsapp-vs-telegram-vs-signal-compWhatsIAiva-cual-es-la-app-de-mensajeria-mas-segura

Sjaak Laan. (n.d.). It Infrastructure Architecture: Infrastructure Building Blocks and Concepts - Sjaak Laan - Google Libros. Retrieved October 4, 2021, from https://books.google.com.co/books?id=wThXV8SQvtcC&printsec=frontcover&hl=es#v=onepage&q=rto&f=false