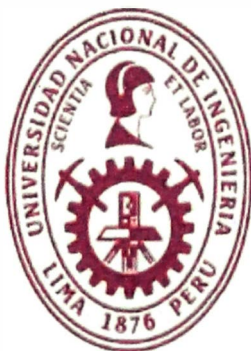


UNIVERSIDAD NACIONAL DE INGENIERIA

FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS



TRABAJO DE SUFICIENCIA PROFESIONAL

**“SISTEMA DE INTERFACES PARA LA ATENCIÓN DE
CLIENTES EN UN CALL CENTER”**

PARA OBTENER EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS

ELABORADO POR:

ROBERTO GERÓNIMO ZÁRATE MENDOZA

ORCID: 0000-0002-0544-5944

ASESOR:

DR. HILARIO ARADIEL CASTAÑEDA

ORCID: 0000-0001-6921-6721

LIMA - PERU

2024

Dedicatoria

Quiero expresar mi más profunda gratitud a mis seres queridos: a mi compañera de vida, Cynthia; a mis progenitores, Mercedes y Gerónimo; y a mi hermano, Edgar David. Su inquebrantable respaldo y aliento han sido fundamentales en este proceso.

Agradecimientos

Expreso mi sincero reconocimiento al cuerpo
docente de la Universidad Nacional de Ingeniería.

Su compromiso y valiosa instrucción han sido pilares
fundamentales durante mi formación académica.

A mi familia que me brindaron ánimo,
motivación y aliento en cada etapa
de este viaje académico

RESUMEN

Esta investigación se ha enfocado en la creación de un conjunto de interfaces interactivas diseñadas para optimizar un área operativa encargada del servicio al cliente en una compañía del sector de las telecomunicaciones, la presente investigación es de tipo aplicada y tiene un enfoque cuantitativo. Se han analizado los efectos del aplicativo de interfaces (variable independiente) sobre la atención al cliente en un call center (variable dependiente). En este análisis, se evaluará si la implementación del aplicativo de interfaces ha producido mejoras en la calidad de la atención al cliente en el call center.

La investigación abarcó el análisis de aproximadamente 562 mil interacciones con clientes, registradas en el transcurso de un trienio comprendido entre 2019 y 2021.

Durante el análisis de resultados se encontró que la variable atención al cliente en un call center en sus dimensiones: eficiencia y eficacia tuvo una mejora considerable al cual se vio afectada por la variable independiente el aplicativo de interfaces.

Palabras clave: base de datos relacional, sistema de interfaces, atención en un call center, java, vpn, operaciones entre tablas.

ABSTRACT

This research has focused on the creation of a set of interactive interfaces designed to optimize an operational area in charge of customer service in a company in the telecommunications sector. This research is applied and has a quantitative approach. The effects of the interface application (independent variable) on customer service in a call center (dependent variable) have been analyzed. In this analysis, it will be evaluated whether the implementation of the interface application has produced improvements in the quality of customer service in the call center.

The investigation covered the analysis of approximately 562 thousand interactions with clients, recorded over the course of a three-year period between 2019 and 2021.

During the analysis of results, it was found that the variable customer service in a call center in its dimensions: efficiency and effectiveness had a considerable improvement, which was affected by the independent variable the application of interfaces.

Keywords: relational database, interface system, service in a call center, java, vpn, operations between tables.

TABLA DE CONTENIDO

<i>Dedicatoria</i>	i
Agradecimientos	ii
RESUMEN	iii
ABSTRACT	iv
TABLA DE CONTENIDO	v
LISTA DE TABLAS	ix
LISTA DE FIGURAS	x
INTRODUCCION	xii
CAPITULO I.....	1
PARTE INTRODUCTORIA DEL TRABAJO.....	1
1.1. Generalidades	2
1.1.1. Organigrama	3
1.1.2. América Móvil Perú SAC	6
1.2. Descripción del problema:	10
1.2.1. Problemática	10
1.2.1.1. Problema General:	11
1.2.1.2. Problema específico:.....	11
1.2.3. América Móvil Perú SAC	12
1.2.4. MISIÓN.....	12
1.2.5. VISIÓN	12
1.2.6. Mapa de Procesos.....	12

1.2.7. Clientes	13
1.2.8. Productos	14
1.2.9. Justificación	14
1.3. Objetivos de Estudio	15
1.3.1. Objetivo General	15
1.3.2. Objetivos Específicos	15
1.4. Antecedentes Investigativos	16
1.4.1. Nacionales	16
CAPITULO II	17
MARCO TEÓRICO Y CONCEPTUAL	17
2.1. Variables	17
2.1.1. Variable Dependiente: Atención al cliente	17
2.1.2. Variable Independiente: Aplicativo de interfaces	19
2.2 Marco Conceptual	20
2.2.1. Base de datos Relacional	20
2.2.2. Modelo Relacional	22
2.2.2.1 Entidad	23
2.2.2.2. Relación	23
2.2.3. Query	23
2.2.3.1. DDL	24
2.2.3.2. DML	30
2.2.2.3. DCL	31
2.2.4. Trigger	32
2.2.5. Vista	32
2.2.6. Procedure y functions	33
2.2.7. Event	35

2.2.8. Accesos a objetos de base de datos	37
2.2.8.1. Grant	37
2.2.8.2. Revoke	40
2.3. Virtual Private Network	41
2.3.1. Intranet y VPN	41
2.3.2. Riesgos	42
2.3.2.1. Firewall	42
2.3.3. Componentes de una VPN	43
2.3.4. Clasificación de VPN	44
2.4. Java	45
2.4.1. JVM	46
2.4.2. JDBC	46
2.4.3. LOG4J	47
2.4.4. Multiplataforma	47
2.4.7. Manejo de errores	48
2.5. Kanban	49
2.5.1. Axioma 1	49
2.5.2. Axioma 2	49
2.6. Linux	50
2.6.1. Scripts	51
2.7. Navicat	51
2.8. Marco SCRUM	52
2.8.1. Pilares de SCRUM	53
2.8.2. Valores de SCRUM	54
2.8.3. Roles de SCRUM	54
2.8.4. Eventos de SCRUM	55

2.9. Diseño de Base de datos.....	58
2.9.1. Fase de diseño de base de datos.....	60
Normalización	67
CAPITULO III	72
DESARROLLO DEL TRABAJO DE INVESTIGACIÓN	72
3.1. Organización	72
3.2. Actividades realizadas	92
3.2.1. FASE DE CONCEPCIÓN.....	92
3.2.2. FASE DE ELABORACIÓN	93
3.2.3. Fase de Construcción	94
3.2.4. Fase de Entrega.....	95
3.2.5. Validaciones	95
CAPITULO IV.....	97
ANÁLISIS Y DISCUSIÓN DE RESULTADOS	97
CONCLUSIONES	101
RECOMENDACIONES.....	102
REFERENCIAS BIBLIOGRÁFICAS.....	103
ANEXOS.....	107
Anexo 1:	107
Anexo 2:	116
Anexo 3:	119
Anexo 4:	125
Anexo 5:	127

LISTA DE TABLAS

Tabla 1. Cronograma de Sprints	9
--------------------------------------	---

LISTA DE FIGURAS

Figura 1. <i>Organigrama por direcciones de América Móvil Perú S.A.C.</i>	4
Figura 2. <i>Organigrama de Servicio al Cliente</i>	5
Figura 3. <i>Organigrama de las Mercado Corporativo</i>	6
Figura 4. <i>Cantidad de trabajadores durante los últimos 12 meses.</i>	9
Figura 5. <i>Proceso de atención de solicitud de orden de trabajo</i>	13
Figura 6. <i>Sintaxis de comando create database.</i>	25
Figura 7. <i>Sintaxis de comando create table</i>	25
Figura 8. <i>Sintaxis completa de comando create table</i>	26
Figura 9. <i>Sintaxis de comando alter table.</i>	28
Figura 10. <i>Sintaxis de comando drop table</i>	30
Figura 11. <i>Sintaxis de comando select.</i>	31
Figura 12. <i>Sintaxis de comando sql create view.</i>	33
Figura 13. <i>Sintaxis del comando de creación de un procedure.</i>	34
Figura 14. <i>Ejecución de un procedure en Mysql.</i>	35
Figura 15. <i>Sintaxis del comando para crear eventos.</i>	36
Figura 16. <i>Sintaxis del comando grant</i>	38
Figura 17. <i>Sintaxis del comando Show</i>	39
Figura 18. <i>Un ejemplo para comando show</i>	40
Figura 19. <i>Comando “show grants for”</i>	40
Figura 20. <i>Sintaxis del comando revoke.</i>	41
Figura 21. <i>Conexión entre un cliente VPN y un servidor VPN.</i>	43
Figura 22. <i>Gráfico de tickets promedios diarios atendidos dentro de las primeras 24 horas por asesor</i>	97
Figura 23. <i>Gráfico de tickets promedios diarios atendidos por asesor.</i>	98
Figura 24. <i>Gráfico de la tasa de Reincidencias por mes</i>	98

Figura 25. *Gráfico de la tasa de Solución Visita no efectiva por mes.99*

INTRODUCCION

En la era actual, los proveedores de servicios de telecomunicaciones se encuentran ante el reto persistente de elevar los estándares de atención a sus usuarios, especialmente en áreas críticas como los centros de atención telefónica, o call centers. La atención eficiente y efectiva al cliente no solo mejora la satisfacción del usuario, sino que también optimiza los procesos internos y contribuye significativamente a la competitividad de la empresa. En este marco, el desarrollo de sistemas de interfaces que faciliten y mejoren la interacción con los clientes se ha convertido en una necesidad imperante.

Esta investigación aborda la creación de un conjunto de interfaces interactivas destinadas a optimizar las operaciones del centro de atención telefónica de América Móvil Perú SAC, una compañía líder en el sector de las telecomunicaciones peruano. La finalidad primordial de esta investigación consiste en conceptualizar y someter a prueba una plataforma innovadora orientada a optimizar la calidad y rapidez en la interacción con los usuarios, mejorando la gestión de tickets y la resolución de problemas. Se busca también analizar el impacto de dicho sistema en la operatividad del call center, considerando métricas clave como la eficacia, eficiencia, y usabilidad del sistema implementado.

Esta investigación tiene como objetivo principal analizar el impacto que ejercen las distintas interfaces de un sistema en la percepción de calidad del servicio por parte de los clientes de una empresa de telecomunicaciones. Para responder a esta pregunta, se han identificado varios problemas específicos relacionados con la efectividad del sistema, su facilidad de uso, y la percepción de los usuarios sobre la mejora en la atención recibida.

La metodología utilizada en este estudio es de tipo aplicada, con un enfoque cuantitativo. Se analizaron datos obtenidos de 562,000 atenciones realizadas entre los años 2019 y 2021. Este análisis permitió establecer correlaciones entre la implementación del sistema de interfaces y los resultados en la calidad de la atención al cliente.

El informe está estructurado en varios capítulos, comenzando con una descripción detallada del contexto de la empresa América Móvil Perú SAC y la problemática que enfrenta en su call center. A continuación, se presentan los detalles del desarrollo del sistema de interfaces, seguido por el análisis de los resultados obtenidos tras su implementación. Finalmente, se discuten las conclusiones y recomendaciones para futuros trabajos en esta área.

CAPITULO I

PARTE INTRODUCTORIA DEL TRABAJO

Según Galitz (2007), una aplicación con interfaces de usuario consta de uno o varios grupos de campos que interactúan con humanos. Estos campos se estudian, planean y diseñan para que la interacción entre los humanos y los sistemas sea efectiva y que las necesidades de las personas respecto al sistema se satisfagan de la manera más eficiente posible. Las interfaces, en esencia, tienen dos grandes grupos de información: entradas y salidas.

Las entradas son los datos que el usuario comunica al sistema a través de dispositivos periféricos como un mouse, teclado, micrófono, capturadora de video u otros sensores de interacción.

Las salidas son los resultados que el sistema muestra a través de alguna interfaz en el computador.

El presente informe presenta en el marco teórico las diferentes herramientas software asociadas al desarrollo de las interfaces mostradas, este informe se ha realizado con las evidencias y cómo el aplicativo de interfaces ha modificado el proceso de atención de tickets en un call Center.

La justificación práctica: el presente informe muestra las historias de usuarios implementadas bajo la metodología SCRUM en donde se

implementará una solución enfocada en desarrollos en base de datos que brindará soporte a la operación de 2 campañas call de center: Atención Tecnológica Fija y Análisis de SOT.

Con estos desarrollos se podrá medir la productividad de los equipos, se conectará con métricas a tiempo real de la operación, se podrá integrar varios sistemas dentro del ecosistema de CLARO y les permitirá a los asesores interactuar con el sistema con una interface amigable y fácil de usar que es el cliente de base de datos MariaDB/Navicat, lo cual permitirá ahorrar horas hombre.

El objetivo general del presente informe es desarrollar un sistema de interfaces que permita brindar atención a los clientes en una empresa de telecomunicaciones.

1.1. GENERALIDADES

Un call center es un área operativa utilizada con el propósito de atender tickets los cuales pueden ser recepción y envío de información a través de un medio de comunicación. Un call center puede manejar llamadas/tickets de servicio al cliente, soporte técnico, telemarketing y otras actividades relacionadas con la atención al cliente.

Desde su creación en la década de 1960, los call centers han evolucionado significativamente. Con el avance de la tecnología, los call centers han adoptado nuevos sistemas de interfaces que permiten una atención más eficiente y personalizada. Hoy en día, las interfaces de usuario juegan un rol importante en la optimización en la atención al cliente.

Mejorar los sistemas con interfaces de forma iterativa en los call centers es importante ya que con eso aumenta la probabilidad de brindar una atención al cliente rápida y efectiva. Un sistema de interfaces bien diseñado puede reducir el tiempo de espera, mejorar la experiencia del cliente y aumentar la efectividad del uso de recursos. Este estudio se centra en el desarrollo de un sistema de interfaces innovador que optimice estos procesos.

Este documento va a analizar el desarrollo de un sistema de interfaces para la atención de clientes en un call center. No se abordan aspectos como la infraestructura física del call center o el entrenamiento del personal, aunque se reconocen como componentes importantes del funcionamiento general.

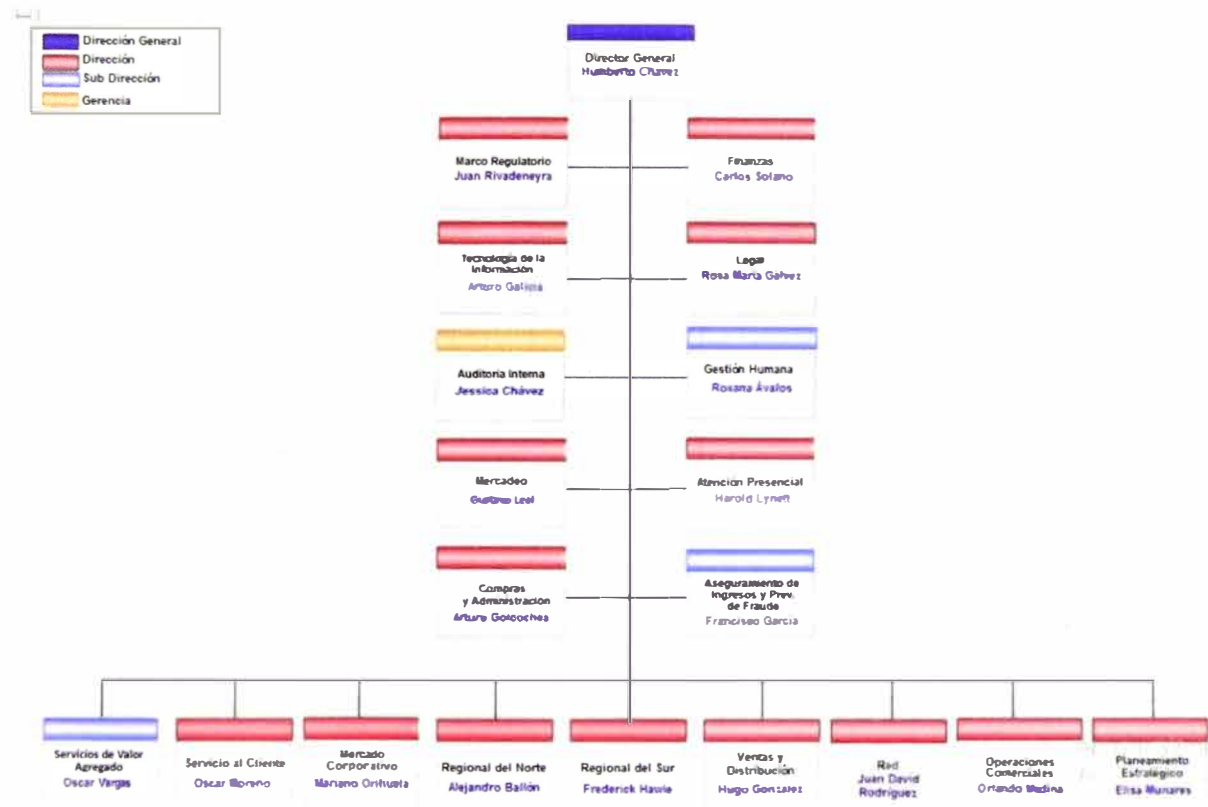
La implementación de un sistema de interfaces mejorado en los call centers tiene un impacto significativo tanto en la experiencia del cliente como en la eficiencia operativa del call center. Este proyecto proporciona una solución teórica y práctica que puede ser adoptada por call centers para mejorar su servicio y rendimiento.

1.1.1. Organigrama

La empresa se organiza en 15 Direcciones, 3 subdirecciones y una gerencia (Figura 1).

Figura 1

Organigrama por direcciones de América Móvil Perú S.A.C.

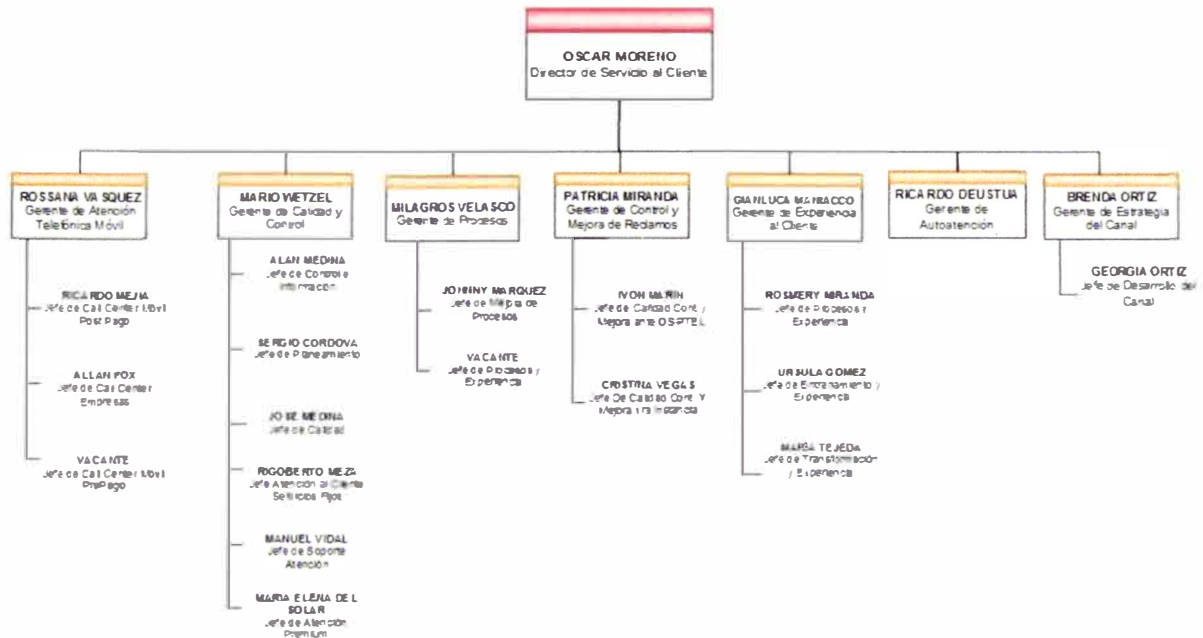


Nota: Documentación Administrativa América Móvil Claro Perú S.A.C.

Dentro de los procesos que se van a revisar en el presente informe es necesario mencionar a 2 direcciones: Servicio al Cliente y Mercado Corporativo, sus respectivos organigramas se presentan en la Figura 2 y la Figura 3.

Figura 2.

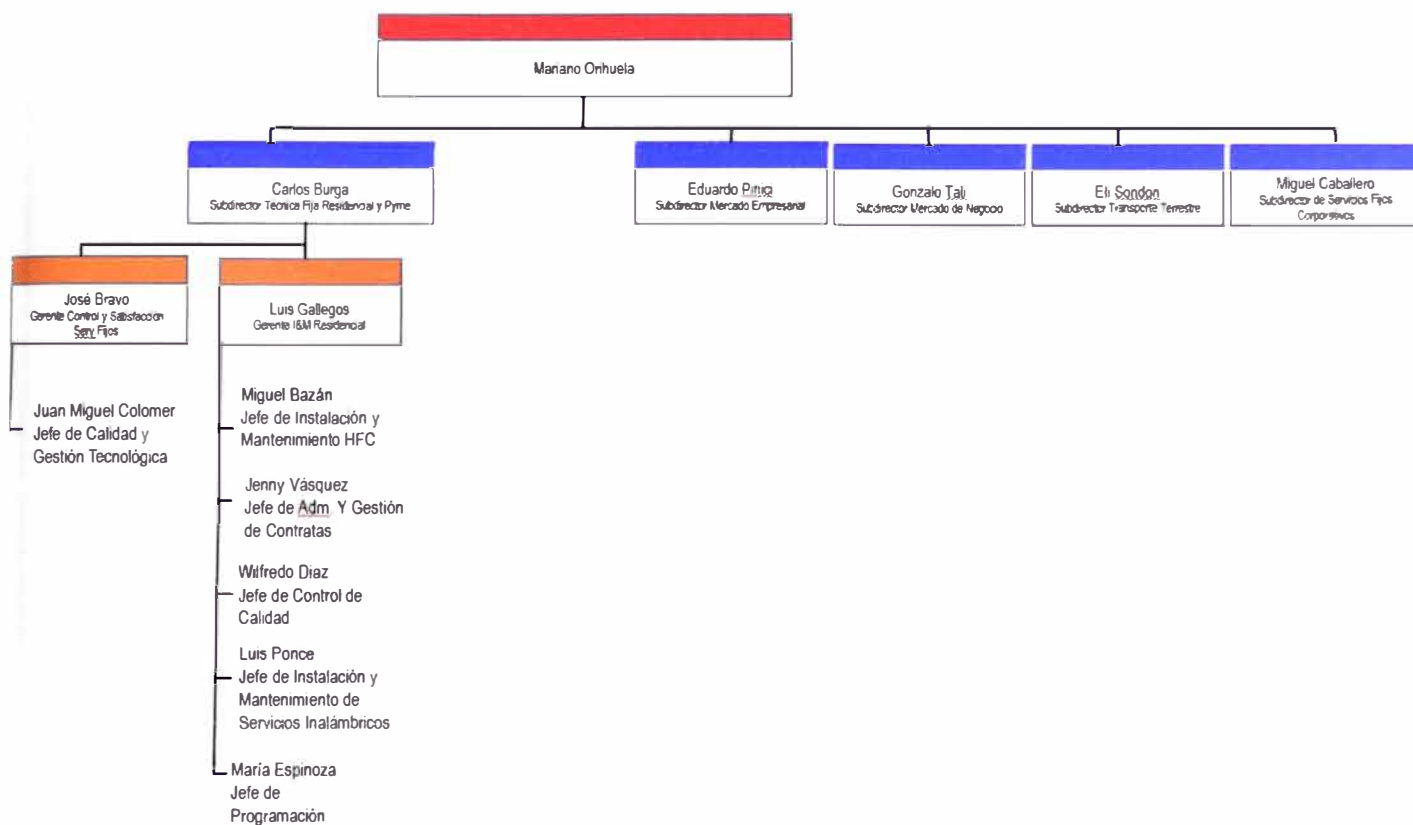
Organigrama de Servicio al Cliente



Nota: Documentación Administrativa América Móvil Claro Perú S.A.C.

Figura 3.

Organigrama del Mercado Corporativo



Nota: Documentación Administrativa América Móvil Claro Perú S.A.C.

1.1.2. América Móvil Perú SAC

América Móvil Perú S.A.C., una compañía líder en el sector de las telecomunicaciones, inició sus operaciones en el territorio peruano en agosto de 2005. Con una sólida experiencia internacional y un profundo conocimiento del mercado local, la empresa se ha consolidado como uno de los principales proveedores de servicios de comunicación, información y entretenimiento del país.

Desde sus inicios, América Móvil Perú se ha destacado por brindar excelentes servicios, ofreciendo una amplia cobertura de red en todo el territorio nacional y brindando a sus usuarios acceso a las tecnologías más innovadoras.

La compañía ha realizado importantes inversiones en infraestructura de telecomunicaciones, la principal finalidad es aumentar la calidad de los servicios y la experiencia de sus clientes además de potenciar el desarrollo digital del país. En línea con esta estrategia, América Móvil Perú ha implementado un ambicioso plan de expansión de su red de fibra óptica, debido a estas mejoras ofrecerá servicios de mayor velocidad y calidad a un número cada vez mayor de usuarios. Asimismo, la empresa ha iniciado nuevos proyectos para fomentar la inclusión digital y reducir la brecha digital en el Perú.

El RUC de la empresa de la cual el presente documento va analizar es: 20467534026, tiene varias sedes desde donde ejecuta sus diferentes procesos operativos en todo el territorio Nacional, su sede Principal Administrativa se encuentra ubicada en la Avenida Arriola 480 La Victoria, Lima.

La empresa tiene en su oferta comercial servicios para pymes, empresas y personas. Dentro de la Oferta de servicios para empresa se encuentran los Servicios Fijos, Móviles y soluciones empresariales.

La oferta de servicios para pymes se conoce como Claro Negocios y esta ofrece servicios fijos, móviles y Cloud. Los principales servicios para personas son: Claro Hogar, Móvil.

Según la SUNAT la cantidad de colaboradores durante el último año es como sigue en la Figura 4

Figura 4

Cantidad de trabajadores durante los últimos 12 meses.

Información de Trabajadores y/o Prestadores de Servicio			
La información mostrada a continuación corresponde a lo declarado por el contribuyente en la Planilla Electrónica o PLAME ante la SUNAT. La información presentada corresponde a los 12 últimos periodos vencidos al mes anterior al día de la consulta.			
Período	Nº de Trabajadores	Nº de Pensionistas	Nº de Prestadores de Servicio
2020-06	3 944	0	20
2020-07	3 923	0	32
2020-08	3 926	0	36
2020-09	3 912	0	34
2020-10	3 911	0	35
2020-11	3 875	0	42
2020-12	3 909	0	54
2021-01	3 873	0	24
2021-02	3 843	0	38
2021-03	6 220	0	34
2021-04	3 786	0	42
2021-05	3 767	0	25

Nota: <http://www.sunat.gob.pe>

1.2. DESCRIPCIÓN DEL PROBLEMA:

En las empresas privadas debido al auge en las automatizaciones de los sistemas actualmente todas las áreas de atención al cliente poseen procesos los cuales se encuentran estandarizados, uno de sus principales problemas son los elevados costos asociados a las plataformas de atención, al tener procesos complejos otro aspecto en donde las empresas necesitan un soporte adecuado es que los sistemas deben soportar debidamente los procesos actuales.

La empresa cuenta con sistemas legados los cuales no se integran fácilmente a los nuevos procesos/sistemas.

Al medir el NPS del proceso de atención de clientes este es bajo e inclusive negativo si consideramos ciertos segmentos específicos, es decir que otro problema que tienen las empresas que atienden masivamente a sus clientes con procesos estandarizados es el bajo NPS.

En el presente trabajo se va a tratar la problemática de la atención los clientes en un área específica asociado a los servicios fijos.

1.2.1. Problemática

Formulación del problema:

El proceso de atención de tickets asociados a la atención al cliente en un área específica en la empresa de telecomunicaciones es una variable de estudio para el presente trabajo, posee las siguientes características:

La atención del cliente se realiza usando sistemas de información inicialmente concebidos para clientes con servicios móviles, debido a esto los sistemas usados en la atención están desactualizadas.

Los sistemas de información actualmente usados para carecen de integración con las plataformas legados, al tenerlas no integradas se traducen en elevados tiempos de atención.

Al no contar con un sistema apropiado para la atención/soporte a los clientes, la eficiencia operativa disminuye ya que la información se encuentra dispersa y no integrada en una sola plataforma.

Debido a los problemas antes mencionados se cuenta un bajo nivel de NPS, es una métrica asociada a la atención de esta área específica de atención al cliente.

1.2.1.1. Problema General:

¿Cómo impacta la implementación de un sistema de interfaces en la eficiencia y eficacia de la atención al cliente en un call center, considerando la cantidad de tickets atendidos, los tiempos de respuesta, la tasa de reincidencias y la tasa de visitas no efectivas?

1.2.1.2. Problema específico:

¿Qué tan efectivo es el sistema de interfaces en incrementar la cantidad de tickets atendidos por asesor, tanto en total como durante las primeras 24 horas después de generar los tickets, en comparación con el sistema anterior?

¿De qué manera influye la implementación del sistema de interfaces en la eficacia de la atención al cliente, medida a través de la tasa de reincidencias mensuales, en comparación con el proceso de atención previo?

¿Cómo afecta el sistema de interfaces la tasa de visitas no efectivas en la atención al cliente, comparando los resultados antes y después de su implementación?

1.2.3. América Móvil Perú SAC

La empresa la cual sustento el presente informe es América Móvil Perú S.A.C., opera en el Perú desde el 10 de agosto del 2005, es una empresa que brinda servicios de telecomunicaciones, datos y entretenimiento.

1.2.4. MISIÓN

Ser la empresa líder en telecomunicaciones, reconocida por la excelencia de nuestros servicios, la innovación constante y nuestro compromiso con la sostenibilidad. Buscamos conectar a las personas, impulsar el crecimiento económico y contribuir al desarrollo social. Para lograrlo, invertimos en la capacitación de nuestros colaboradores, ampliamos nuestra cobertura de red y desarrollamos soluciones tecnológicas que agreguen valor a nuestros clientes.

1.2.5. VISIÓN

Ser la empresa líder en telecomunicaciones en el Perú.

1.2.6. Mapa de Procesos.

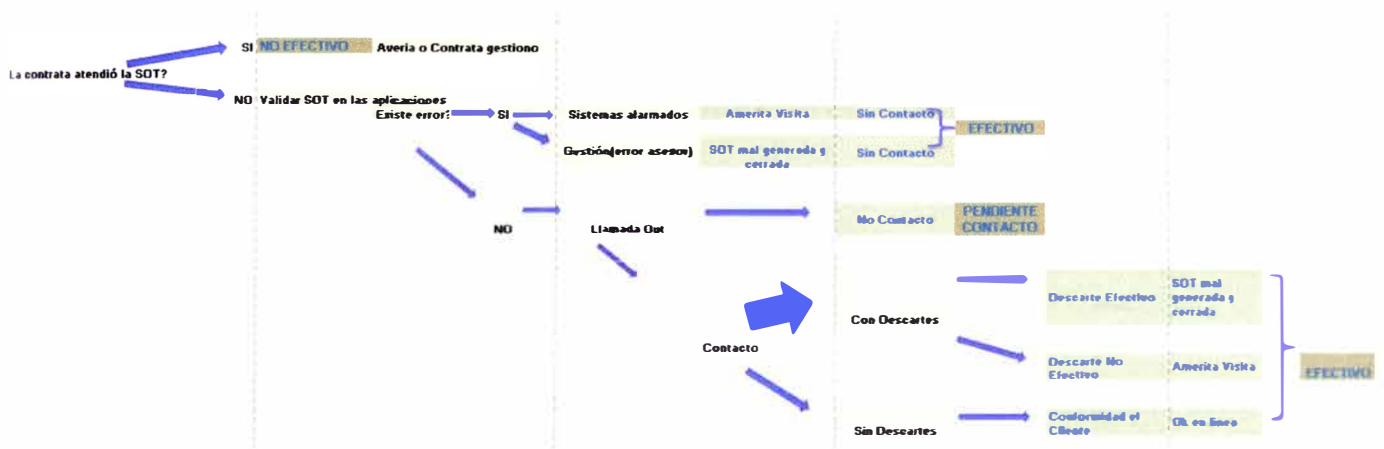
La empresa cuenta con procesos en todas sus gerencias enfocados al cumplimiento de sus respectivos objetivos, en el presente informe se va a

detallar los requerimientos asociados a 2 procesos de atención al cliente en la gerencia de Satisfacción y Control de Servicios Fijos.

A continuación se muestra el flujograma del proceso de atención de solicitudes de orden de trabajo en la Figura 5.

Figura 5.

Proceso de atención de solicitud de orden de trabajo



Nota: Documentación Administrativa América Móvil Claro Perú S.A.C.

1.2.7. Clientes

Los principales clientes son:

Minera Cerro Verde

Municipalidad Provincial de Tacna

Minera las Bambas S.A.

Compañía de Seguridad Prosegur S.A.

Universidad Peruana de Ciencias Aplicadas

Municipalidad Provincial de Piura

Pontificia Universidad Católica del Perú

1.2.8. Productos

Los productos que ofrece en sus respectivos catálogos de servicios se encuentran en las siguientes categorías:

- a) Pymes
- b) Personas
- c) Corporativos.

En todas las categorías tienen los productos divididos en:

- Servicios Móviles
- Servicios Fijos
- Servicios Cloud

Con estos productos la empresa cubre las necesidades de conectividad de los diferentes segmentos de mercado.

1.2.9. Justificación

El presente informe muestra las historias de usuarios implementadas bajo la metodología SCRUM en donde se implementará una solución enfocada en desarrollos en base de datos que brindará soporte a la operación de 2 campañas call de center: Atención Tecnológica Fija y Análisis de SOT.

Con estos desarrollos se podrá medir la productividad de los equipos, se conectará con métricas a tiempo real de la operación, se podrá integrar varios sistemas dentro del ecosistema de CLARO y les permitirá a los asesores interactuar con el sistema con una interface amigable que es el cliente de MariaDB, Navicat, lo cual permitirá ahorrar horas hombre.

Con una sólida trayectoria internacional y un profundo conocimiento del mercado, la empresa ha logrado permanecer líder en el sector de las telecomunicaciones peruano. Desde sus inicios, se ha comprometido a ofrecer servicios de excelencia y una amplia cobertura a nivel nacional, brindando oportunidades de crecimiento a través de la conectividad. Su plan de inversiones se centra en el despliegue sostenible de una robusta infraestructura de telecomunicaciones busca ampliar la experiencia del cliente, potenciar la red 4.5G y mejorar la cobertura de la fibra óptica. De esta manera, aporta al desarrollo de las comunicaciones del país.

1.3. OBJETIVOS DE ESTUDIO

1.3.1. Objetivo General

Evaluar el impacto de la implementación de un sistema de interfaces en la mejora de la eficiencia y eficacia de la atención al cliente en un Call center.

1.3.2. Objetivos Específicos

Objetivo Específico 1: Cuantificar el incremento en la cantidad de tickets atendidos por asesor, tanto en total como durante las primeras 24 horas luego de generar los tickets, antes y después de la implementación del sistema de interfaces

Objetivo Específico 2: Analizar la influencia del sistema de interfaces en la eficacia de la atención al cliente, considerando la tasa de reincidencias por mes, antes y después de la implementación del sistema de interfaces.

Objetivo Específico 3: Cuantificar el incremento de la tasa de visitas no efectiva de los tickets atendidos por el asesor antes y después de la implementación del sistema de interfaces.

1.4. ANTECEDENTES INVESTIGATIVOS

1.4.1. Nacionales

Según Miranda Torres y Blax Arteaga (2018) en su trabajo de investigación plantean el desarrollo de un sistema web mostrando el diseño del sistema en la notación UML usando el RUP(Proceso unificado de Rational) en la facultad de Ingeniería de sistemas y civil en la Universidad Nacional de Ucayali, los autores nos muestran que antes de la implementación del sistema el registro de trámite documentario tomaba 6 minutos con 13 segundos en promedio y luego de la implementación del sistema web el registro de un trámite documentario tomaba en promedio 30 segundos, se observa un ahorro de recursos con respecto al tiempo empleado en el registro del trámite, con este resultado nos indica que en caso los responsables de implementación de este sistema web deben tener en cuenta como un backup de base de datos, así también como nodos de respaldo ante caídas y que el sistema podría estar dentro de la infraestructura de la universidad pero también se debe considerar que con las nuevas tendencias se podría llevar a implementación en la nube.

CAPITULO II

MARCO TEÓRICO Y CONCEPTUAL

2.1. VARIABLES

2.1.1. Variable Dependiente: Atención al cliente

Según Gil López (2019) considera la atención al cliente como un conjunto de actividades que realiza la organización identificando las necesidades básicas de sus clientes, buscando cumplir con las expectativas del clientes busca la satisfacción del cliente como objetivo principal y también menciona que la fidelización es muy importante en la relación con el cliente ya que mide si se ha logrado la repetición de la compra, según Muther (2002) la atención al cliente es la relación que se establece entre la empresa y el cliente, estas relaciones pueden organizarse en 4 partes: adquisición de clientes, retención de clientes antiguos, innovación de productos, mantenimiento de productos actuales. Cada parte posee diferentes informaciones que se debe tratar con los clientes y maneja una forma de comunicación distinta dependiendo las estrategias que use la organización para mantener la relación con los clientes. Muther (2002) nos indica que es indispensable usar elementos informáticos tales como sistemas de información que la permita a la organización tener

procesos estandarizados y poder brindar una correcta atención en los 4 procesos antes mencionados.

Dimensiones:

- Eficiencia:

Según Arenal Laza (2019) la eficiencia en un proceso de atención a clientes significa alcanzar el objetivo definido con la menor cantidad recursos disponibles, es decir para poder medir el proceso de atención se debe contar con métricas definidas, en caso se cumplan las metas a través de objetivos con el uso optimizado de los recursos asignados.

En la presente dimensión he considerado los siguientes indicadores:

1. Cantidad de tickets promedios diarios atendidos dentro de las primeras 24 horas por asesor: esta métrica está relacionada a un objetivo de la campaña y mide la eficacia del proceso.
2. Cantidad de tickets promedios diarios atendidos por asesor: el aplicativo brinda soporte al asesor a realizar las tareas encomendadas, este aplicativo le ayuda al asesor a cumplir su cuota de tickets diario.

- Eficacia:

Según Jiménez Paneque (2004) y De La Cruz Fernández (2017) la eficacia en un proceso de atención a clientes, es el cumplimiento de objetivos organizacionales, mientras se cumplan en mayor medida los procedimientos y tareas que permitan alcanzar estos los procesos son más eficaces.

Para la presente dimensión he considerado los siguientes indicadores:

1. Tasa de Reincidencias por mes: en la operación los clientes vuelven a contactar con el call center por diversos problemas, el equipo de atención de tickets debe asegurarse de revisar todas las plataformas y verificar problemas no reportados por el cliente.
2. Tasa de Solución Visita no efectiva por mes. Después de que un ticket sea atendido el asesor puede etiquetarlo para que sea atendido por una cuadrilla en campo y en caso que esta cuadrilla culmine esta atención, toda ticket atendido por una cuadrilla será tipificado es decir tendrá un motivo de resolución, estos motivos se agrupan para poder realizar análisis de las atenciones, estos tickets revisados por el call center y atendidos por una cuadrilla no deberían tipificarse como visita no efectiva que indica que el problema se pudo solucionar sin necesidad de atención de la cuadrilla en campo.

2.1.2. Variable Independiente: Aplicativo de interfaces

Según Galitz (2007) un aplicativo con interfaces de usuario es uno o varios grupos de campos los cuales interactúan con humanos, estos campos se estudian, son planeados y diseñados para que los humanos y los sistemas interactúen de una buena forma y que las necesidades de las personas con respecto al sistema sean satisfechas a través de la forma más óptima, se debe considerar las limitaciones del sistema, las limitaciones de los usuarios, cómo el sistema trabaja y como las personas interactúan con el sistema, las interfaces básicamente tienen 2 grandes grupos de información: entradas y salidas.

Las interfaces del aplicativo forman parte del software del aplicativo, es la parte que los usuarios pueden oír, ver e interactuar del sistema.

Las entradas son los datos que el usuario comunica al sistema a través de un dispositivo periférico como un mouse, teclado, audio, capturadora de video u otros sensores de interacción.

Las salidas son los resultados que muestra el sistema a través de alguna interface en el computador.

Dimensiones:

- Usabilidad:

Según Galitz (2007) la usabilidad de un aplicativo de interfaces son las cualidades o atributos que evalúan la facilidad con la cual se usa una interface de usuario, también se refiere a los métodos de mejora del proceso de diseño de todo o parte del aplicativo.

Para la presente dimensión he considerado los siguientes indicadores:

1. Tiempo acceso a la información
2. Tasa de errores de usuario.

2.2 MARCO CONCEPTUAL

2.2.1. Base de datos Relacional

Según Piñeiro Gómez (2014) las bases de datos se empezaron a usar en la década del 70, antes se usaban ficheros, existían 2 tipos de memoria la RAM y la secundaria, la RAM es llamada memoria volátil, en los ficheros se

organizaban toda la información que se requería, por ejemplo si necesitaba guardar la información de un cliente, en un registro se guardaba su información de su nombre, apellido, localidad, provincia, email y otros, él menciona que uno de los mayores beneficios es el valor informativo quiere decir que la suma de tener el conjunto de datos en un repositorio es mayor a tenerlo de forma individual, él también indicó que una de las desventajas que más afecta a las empresas son los altos costos de implementación ya que se requiere un personal calificado para implementar/administrar un sistema de base de datos.

Según Pisco Gómez et al. (2017) los Sistemas de base de datos tienen por objetivo proporcionar acceso múltiple y simultáneo a los usuarios, puede darse el caso que dentro de una misma empresa diferentes grupos organizacionales puedan tener diferentes enfoques y usen los mismos datos. Estos análisis pueden usar la información de manera distinta, pero usarán los mismos datos.

También menciona que los sistemas de Gestión bases de datos son diseñadas para servir propósitos específicos, puede ser almacenar un gran volumen de información, mantener los datos relacionados, u otros motivos, estos datos en cualquier momento sean consultados deben garantizar la no contradicción de los mismos.

Las bases de datos deben contar con alguna forma de seguridad es decir que sólo puedan acceder los usuarios a los objetos que tienen acceso configurado. Los sistemas de gestión de base tienen como característica principal la de poder tener datos relacionados y tener el control centralizado, es decir es que

debe haber un sistema único que permita esta administración, esta función debe recaer en una unidad organizacional que administre la base de datos, dentro de sus funciones estas deben ser:

Validar la creación de objetos en la base de datos.

Garantizar la disponibilidad del servicio

Ser responsable de la redundancia de datos.

Revisar estrategias para mantener el performance en la base de datos.

2.2.2. Modelo Relacional

Según Adiego Rodríguez & Llanos Ferraris (2010) indican que Edgar Frank Codd mencionó en un artículo en 1970 que tenía por título: “Relational model of Data for Large Shared Data Banks”, si buscamos una traducción que considere el contexto de la investigación sería: “un modelo de datos relacional para grandes bancos de datos compartidos”, indicó que se podía realizar una representación lógica de datos y las relaciones entre ellos, llamó a cada fila el nombre de “tupla”, este objeto poseía la siguiente estructura la cual permite almacenar los datos, tiene 2 elementos:

- Campo: Es el nombre de la columna, debe tener un nombre único es decir no debe haber 2 columnas con el mismo nombre.
- Registro: Se refiere a la tupla/fila.

Cada tabla posee un nombre único, dependiendo el motor, éste podría tener distintos seudónimos o alias, la estructura de la tabla se puede expresar como

un archivo plano en donde se definen los tipos de datos de las columnas, su nombre y sus características especiales.

2.2.2.1 Entidad

Según Osorio Rivera (2008) en el modelo Entidad relación se definen las entidades cómo los objetos que pueden distinguirse de otros, estos pueden ser de 2 tipos: concretos y abstractos, una agrupación de entidades se le conoce como un conjunto de entidades.

Una entidad se representa como un conjunto de atributos, para cada atributo existe tipos de datos permitidos o rangos asociados estos conjuntos de valores se le conoce como el Dominio del atributo

2.2.2.2. Relación

Según Osorio Rivera (2008) las relaciones son asociaciones entre varias entidades, esta asociación se manifiesta y posee un algún significado que se le otorgue, mayormente las relaciones se usan como representaciones de la realidad en donde buscamos representar un sistema de tal forma que nos permita guardar, operar y consultar los datos que se necesiten.

2.2.3. Query

Según Mora Rioja (2014) en 1974 Chamberlin y Raymond F. definieron un lenguaje para interactuar con el sistema de gestión de base de datos y por su semejanza al idioma ingles lo denominaron SEQUEL y éste debido a problemas de copyright se cambió a SQL.

SQL, que significa Lenguaje de Consulta Estructurado, se deriva del álgebra relacional y permite gestionar bases de datos mediante acciones como actualizar, añadir o borrar registros.

Las sentencias SQL suelen dividirse en 3 categorías:

- DDL: Lenguaje de definición de datos permite crear estructuras
- DML: Lenguaje de manipulación de datos, permite borrar, escribir, consultar y modificar datos.
- DCL: Lenguaje de control de datos, este lenguaje se encarga de administrar los accesos a los diferentes objetos de base de datos de los usuarios y perfiles.

2.2.3.1. DDL

Según Mora Rioja (2014) los comandos DDL están asociados con la creación, eliminación y modificación de los siguientes objetos de base de datos

- Data base
- View
- Table
- Índice

En la base Mysql al crear los objetos antes mencionados se usa el comando: Create.

La sintaxis del comando **create** se muestran en la Figura 6, Figura 7 y Figura 8 respectivamente.

La sintaxis del comando **alter table** se muestra en la Figura 9.

Figura 6

Sintaxis de comando create database.

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
    [create_option] ...

create_option: [DEFAULT] {
    CHARACTER SET [=] charset_name
    | COLLATE [=] collation_name
}
```

Nota: <https://dev.mysql.com/doc/refman/5.6/en/create-database.html>

Figura 7

Sintaxis de comando create table

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
    [create_option] ...

create_option: [DEFAULT] {
    CHARACTER SET [=] charset_name
    | COLLATE [=] collation_name
}
```

Nota: <https://dev.mysql.com/doc/refman/5.6/en/create-table.html>

Figura 8

Sintaxis completa de comando create table

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    create_definition
    table_options
    partition_options;

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    create_definition
    table_options
    partition_options
    IGNORE | REPLACE
    AS query_expression;

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    ( [tbl_old_tbl_name] [tbl_old_tbl_name] );

create_definition: {
    col_name column_definition
    | (INDEX | KEY) index_name index_type
    key_part
    index_option
    | (FULLTEXT | SPATIAL) INDEX | KEY
    index_name key_part
    index_option
    | CONSTRAINT symbol PRIMARY KEY
    index_type key_part
    index_option
    | CONSTRAINT symbol UNIQUE INDEX | KEY
    index_name index_type key_part
    index_option
    | CONSTRAINT symbol FOREIGN KEY
    index_name col_name
    reference_definition
    CHECK expr
}

column_definition:
    data_type [NOT NULL | NULL | DEFAULT]
    default_value
    AUTO_INCREMENT | UNIQUE | KEY | PRIMARY;

KEY:
    COMMENT 'string'
    COLLATE collation_name
    COLUMN_FORMAT {FIXED | DYNAMIC | DEFAULT}
    STORAGE {DISK | MEMORY}
    reference_definition

data_type:
    see Chapter 11: Data Types;

key_part:
    col_name [length] [ASC | DESC];

index_type:
    USING {BTREE | HASH};

index_option: {
    KEY_BLOCK_SIZE [=] value
    | index_type
    | WITH PARSER parser_name
    | COMMENT 'string'
}

reference_definition:
    REFERENCES tbl_name key_part
    MATCH FULL | MATCH PARTIAL | MATCH SIMPLE
    ON DELETE reference_option
    ON UPDATE reference_option;

reference_option:
    RESTRICT | CASCADE | SET NULL | NO ACTION |
    SET DEFAULT;
```



```

table_options:
    table_option [...] table_option

table_option: {
    AUTO_INCREMENT [-] value
    AVG_ROW_LENGTH [-] value
    [DEFAULT] CHARACTER SET [-] charset_name
    CHECKSUM [-] {0 | 1}
    [DEFAULT] COLLATE [-] collation_name
    COMMENT [-] 'string'
    CONNECTION [-] 'connect_string'
    {DATA | INDEX} DIRECTORY [-] 'absolute path to
directory'

    DELAY_KEY_WRITE [-] {0 | 1}
    ENGINE [-] engine_name
    INSERT_METHOD [-] { NO | FIRST | LAST }
    KEY_BLOCK_SIZE [-] value
    MAX_ROWS [-] value
    MIN_ROWS [-] value
    PACK_KEYS [-] {0 | 1 | DEFAULT}
    PASSWORD [-] 'string'
    ROW_FORMAT [-] {DEFAULT | DYNAMIC | FIXED |
COMPRESSED | REDUNDANT | COMPACT}
    STATS_AUTO_RECALC [-] {DEFAULT | 0 | 1}
    STATS_PERSISTENT [-] {DEFAULT | 0 | 1}
    STATS_SAMPLE_PAGES [-] value

    TABLESPACE tablespace_name [STORAGE {DISK | MEMORY}]
    UNION [-] tbl_name [tbl_name [...] ]
}

partition_options:
    PARTITION BY
        { [LINEAR] HASH(expr)
        | [LINEAR] KEY ALGORITHM={1 | 2} column_list
        | RANGE(expr) COLUMNS(column_list)
        | LIST(expr) COLUMNS(column_list) }
    [PARTITIONS num]
    SUBPARTITION BY
        { [LINEAR] HASH(expr)
        | [LINEAR] KEY ALGORITHM={1 | 2} column_list
        }
    [SUBPARTITIONS num]
    partition_definition [, partition_definition
    ]

partition_definition:
    PARTITION partition_name
    VALUES
        { LESS THAN { expr | value_list | MAXVALUE }
        | IN (value_list) }
    [ [STORAGE | ENGINE] [-] engine_name
    | COMMENT [-] 'string'
    | DATA DIRECTORY [-] 'data_dir'
    | INDEX DIRECTORY [-] 'index_dir'
    | MAX_ROWS [-] max_number_of_rows
    | MIN_ROWS [-] min_number_of_rows
    | TABLESPACE [-] tablespace_name
    | NODEGROUP [-] node_group_id
    ]
    subpartition_definition
    [...]

subpartition_definition:
    SUBPARTITION logical_name
    [ [STORAGE | ENGINE] [-] engine_name
    | COMMENT [-] 'string'
    | DATA DIRECTORY [-] 'data_dir'
    | INDEX DIRECTORY [-] 'index_dir'
    | MAX_ROWS [-] max_number_of_rows
    | MIN_ROWS [-] min_number_of_rows
    | TABLESPACE [-] tablespace_name
    | NODEGROUP [-] node_group_id
    ]

```

Nota: <https://dev.mysql.com/doc/refman/5.6/en/create-table.html>

La sintaxis del comando Alter, para la base Mysql es como sigue:

Figura No.9

Sintaxis de comando alter table.

```

ALTER [ONLINE | OFFLINE | IGNORE] TABLE
tbl_name
[alter_option [, alter_option] ...]
[partition_options]

alter_option: {
    table_options
  | ADD [COLUMN] col_name column_definition
    [FIRST | AFTER col_name]
  | ADD [COLUMN] (col_name
column_definition, ...)
  | ADD {INDEX | KEY} [index_name]
    [index_type] (key_part, ...)
[index_option] ...
  | ADD {FULLTEXT | SPATIAL} [INDEX | KEY]
[index_name]
    (key_part, ...) [index_option] ...
  | ADD [CONSTRAINT symbol] PRIMARY KEY
    [index_type] (key_part, ...)
    [index_option] ...
  | ADD [CONSTRAINT symbol] UNIQUE [INDEX |
KEY]
    [index_name] [index_type]
    (key_part, ...)
    [index_option] ...
  | ADD [CONSTRAINT symbol] FOREIGN KEY
    [index_name] (col_name, ...)
    reference_definition
  | ADD CHECK (expr)
  | ALGORITHM [=] {DEFAULT | INPLACE | COPY}
  | ALTER [COLUMN] col_name {
    SET DEFAULT {literal | (expr)}
  | DROP DEFAULT
  }
  | CHANGE [COLUMN] old_col_name new_col_name
column_definition
    [FIRST | AFTER col_name]
    [DEFAULT CHARACTER SET [=] charset_name
[COLLATE [=] collation_name]
  | CONVERT TO CHARACTER SET charset_name
[COLLATE collation_name]
  | {DISABLE | ENABLE} KEYS
  | {DISCARD | IMPORT} TABLESPACE
  | DROP [COLUMN] col_name
  | DROP {INDEX | KEY} index_name
  | DROP PRIMARY KEY
  | DROP FOREIGN KEY fk_symbol
  | FORCE
  | LOCK [=] {DEFAULT | NONE | SHARED |
EXCLUSIVE}
  | MODIFY [COLUMN] col_name column_definition
    [FIRST | AFTER col_name]
  | ORDER BY col_name [, col_name] ...
  | RENAME [TO | AS] new_tbl_name
}

```

```

ALTER [ONLINE | OFFLINE] [IGNORE] TABLE
tbl_name
    [alter_option [, alter_option] ...]
    [partition_options]

alter_option: {
    table_options
    | ADD [COLUMN] col_name column_definition
      [FIRST | AFTER col_name]
    | ADD [COLUMN] col_name
      column_definition ...
    | ADD {INDEX | KEY} index_name
      [index_type] [key_part ...]
    [index_option] ...
    | ADD {FULLTEXT | SPATIAL} {INDEX | KEY}
      index_name
      [key_part ...] [index_option] ...
    | ADD [CONSTRAINT symbol] PRIMARY KEY
      [index_type] [key_part ...]
      [index_option] ...
    | ADD [CONSTRAINT symbol] UNIQUE INDEX |
    KEY
      [index_name] [index_type]
      [key_part ...]
      [index_option] ...
    | ADD [CONSTRAINT symbol] FOREIGN KEY
      [index_name] (col_name ...)
      reference_definition
    | ADD CHECK (expr)
    | ALGORITHM [=] {DEFAULT | INPLACE | COPY}
    | ALTER [COLUMN] col_name {
      SET DEFAULT {literal | expr}
      | DROP DEFAULT
    }
    | CHANGE [COLUMN] old_col_name new_col_name
      column_definition
      [FIRST | AFTER col_name]
    | [DEFAULT] CHARACTER SET [=] charset_name
    | COLLATE [=] collation_name
    | CONVERT TO CHARACTER SET charset_name
    | COLLATE collation_name
    | {DISABLE | ENABLE} KEYS
    | {DISCARD | IMPORT} TABLESPACE
    | DROP [COLUMN] col_name
    | DROP {INDEX | KEY} index_name
    | DROP PRIMARY KEY
    | DROP FOREIGN KEY fk_symbol
    | FORCE
    | LOCK [=] {DEFAULT | NONE | SHARED |
    EXCLUSIVE}
    | MODIFY [COLUMN] col_name column_definition
      [FIRST | AFTER col_name]
    | ORDER BY col_name [, col_name] ...
    | RENAME [TO | AS] new_tbl_name
}

```

Nota: <https://dev.mysql.com/doc/refman/5.6/en/alter-table.html>

Para eliminar los objetos antes mencionados se usa el comando: Drop.

La sintaxis del comando Drop para Mysql versión 5.6 se muestra en la Figura 10.

Figura 10

Sintaxis de comando drop table

```
DROP [TEMPORARY] TABLE [IF EXISTS]
    tbl_name [, tbl_name] ...
    [RESTRICT | CASCADE]
```

Nota: <https://dev.mysql.com/doc/refman/5.6/en/alter-table.html>

2.2.3.2. DML

Las sentencias DML se refieren a las instrucciones que se ejecutan en la base de datos modificando la información, se conoce como: data manipulation language. Según Mora Rioja (2014) el comando “select” en sus diferentes variaciones permite mostrar los datos asociados a uno o más objetos de la base de datos.

La sintaxis del comando select se muestra en la Figura 11.

Figura 11

Sintaxis de comando select.

```
SELECT
  [ALL | DISTINCT | DISTINCTROW ]
  [HIGH_PRIORITY]
  [STRAIGHT_JOIN]
  [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
  [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
  select_expr [, select_expr] ...
  [into_option]
  [FROM table_references
    [PARTITION partition_list]]
  [WHERE where_condition]
  [GROUP BY {col_name | expr | position}
    [ASC | DESC], ... [WITH ROLLUP]]
  [HAVING where_condition]
  [ORDER BY {col_name | expr | position}
    [ASC | DESC], ...]
  [LIMIT {[offset,] row_count | row_count OFFSET offset}]
  [PROCEDURE procedure_name(argument_list)]
  [into_option]
  [FOR UPDATE | LOCK IN SHARE MODE]

into_option: {
  INTO OUTFILE 'file_name'
    [CHARACTER SET charset_name]
    export_options
  | INTO DUMPFILE 'file_name'
  | INTO var_name [, var_name] ...
}
```

Nota: <https://dev.mysql.com/doc/refman/5.6/en/select.html>

2.2.2.3. DCL

Las sentencias DCL (Data control language) según Mora Rioja (2014) son las que actualizan los accesos de los usuarios sobre los objetos de base de datos.

Describe la sintaxis y el uso de los comandos Grant y Revoke en los puntos:

2.1.7.1. y 2.1.7.2. del presente trabajo.

2.2.4. Trigger

Según Mora Rioja (2014) Como su nombre lo indica es un “disparador”, se refiere a un programa o aun bloque de código, sentencias con orden lógico que cumplen una función, el cual actualiza algunos datos asociados de uno o más objetos de la base de datos y que se ejecuta antes o después de los siguientes eventos:

- Inserción de un registro.
- Modificación de un registro.
- Eliminación de un registro.

Estos programas en caso se ejecuten después de los eventos mencionados, dentro del bloque de código se pueden usar los valores antes y después del evento.

Una consideración para crear un disparador es la siguiente:

No crear un bucle infinito. Por ejemplo, si desarrollamos 2 triggers para 2 tablas A y B en donde las mismas poseen triggers y en ambos triggers tienen bloques de código en donde ejecutan acciones en donde se ejecutarán ellos mismos, de esta forma se genera bucles infinitos los cuales deben ser evitados.

2.2.5. Vista

Según Mora Rioja (2014) Las vistas son consultas sobre una o más tablas, para el usuario una vista es como una tabla más, brinda una capa más al nivel de seguridad, al ser una consulta guardada en la base de datos tiene mayor eficiencia que una consulta común.

Las vistas no permiten recibir parámetros, es decir solo pueden mostrar valores de campos concretos, una vista puedes mostrar una lista de clientes u otra información, además permite filtrar determinados valores para un determinado dato variable.

Uno de sus principales usos se da en la reportería, otros usos comunes son cuando en un sistema se requiere realizar la misma consulta de manera frecuente sobre un grupo de tablas.

Para crear una vista en Mysql se debe seguir usar el comando sql create view como se muestra en la Figura 12.

Figura 12

Sintaxis de comando sql create view.

```
1 CREATE
2     [OR REPLACE]
3     [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
4     [DEFINER = user]
5     [SQL SECURITY { DEFINER | INVOKER }]
6     VIEW view_name [(column_list)]
7     AS select_statement
8     [WITH [CASCADED | LOCAL] CHECK OPTION]
```

Nota: Fuente: <https://dev.mysql.com>

Se debe tener algunas consideraciones antes de crear una vista:

- No todos los valores de los campos de la vista se pueden actualizar.
- No se puede crear vistas anidando subconsultas.

2.2.6. Procedure y functions

Según Mora Rioja (2014) Los procedures y las functions son conjuntos de instrucciones que se ejecutan en la base de datos, cuando el desarrollador lo

necesite, en ambos casos se ejecutan las líneas de código de forma secuencial.

Los procedures se diferencian de las functions en que estos pueden tener como salida varios sets de datos, mientras que las functions sólo pueden tener como salida un valor.

Para crear un procedure y una function se usa el comando create y para poder modificarlo el comando alter.

Una buena práctica es anteponer el prefijo “P_” al nombrar un procedimiento almacenado y el prefijo “F_” al nombrar una función. La sintaxis del comando de creación de un procedure se muestra en la Figura 13.

Figura 13

Sintaxis del comando de creación de un procedure.

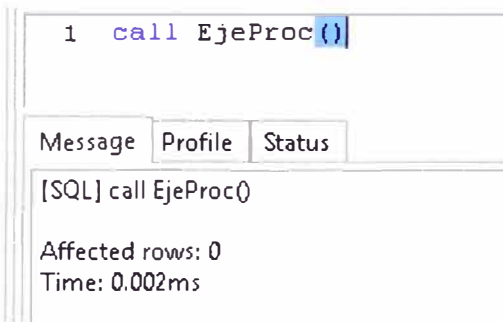
```
1 CREATE
2     [DEFINER = user]
3     PROCEDURE sp_name ([proc_parameter[,...]])
4     [characteristic ...] routine_body
5
6 CREATE
7     [DEFINER = user]
8     FUNCTION sp_name ([func_parameter[,...]])
9     RETURNS type
10    [characteristic ...] routine_body
11
12 proc_parameter:
13     [ IN | OUT | INOUT ] param_name type
14
15 func_parameter:
16     param_name type
17
18 type:
19     Any valid MySQL data type
20
21 characteristic: {
22     COMMENT 'string'
23     | LANGUAGE SQL
24     | [NOT] DETERMINISTIC
25     | { CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }
26     | SQL SECURITY { DEFINER | INVOKER }
27 }
28
29 routine_body:
30     Valid SQL routine statement
```

Nota: Fuente: <https://dev.mysql.com>

Cuando se desee ejecutar un procedure se ejecutará usando el comando call como se muestra en la Figura 14.

Figura 14

Ejecución de un procedure en Mysql.



Nota: Elaboración propia

2.2.7. Event

Según (Murach, 2019) Los eventos son programas que se ejecutan una sola vez o periódicamente en la base de datos, se usan para ejecutar cierta lógica en la base de datos.

Los eventos son usados para realizar labores de consistencia, reportería, o lógica de negocio, este bloque de código verificará la consistencia de información de ciertos campos, también puede sumarizar ciertos ítems en un sistema de logística o de inventario es decir puede precalcular reportes a fin de no procesar un gran volumen de datos lo cual demandaría gran uso de tiempo y recursos.

La sintaxis del comando para crear procedures en Mysql es como se muestra en la Figura 15.

Figura 15

Sintaxis del comando para crear eventos.

```
1 CREATE
2   [DEFINER = user]
3   EVENT
4   [IF NOT EXISTS]
5   event_name
6   ON SCHEDULE schedule
7   [ON COMPLETION [NOT] PRESERVE]
8   [ENABLE | DISABLE | DISABLE ON SLAVE]
9   [COMMENT 'string']
10  DO event_body;
11
12  schedule: {
13    AT timestamp [+ INTERVAL interval] [, ...]
14    | EVERY interval
15    | STARTS timestamp [+ INTERVAL interval] [, ...]
16    | ENDS timestamp [+ INTERVAL interval] [, ...]
17  }
18
19  interval:
20    quantity {YEAR | QUARTER | MONTH | DAY | HOUR | MINUTE |
21              WEEK | SECOND | YEAR_MONTH | DAY_HOUR | DAY_MINUTE |
22              DAY_SECOND | HOUR_MINUTE | HOUR_SECOND | MINUTE_SECOND}
```

Nota: www.mysql.org

Para activar los eventos en una base Mysql o MariaDb, hay 2 formas:

- Modificar la línea de código del archivo de configuración con el siguiente comando:

`event_scheduler=DISABLED`

- Ejecutar cualquiera de los siguientes comandos cuando se tiene el motor Mysql prendido:

El siguiente comando activa los eventos:

`SET GLOBAL event_scheduler = ON`

El siguiente comando activos los eventos en la sesión del usuario:

`SET @@GLOBAL.event_scheduler = ON`

`SET GLOBAL event_scheduler = 1`

`SET @@GLOBAL.event_scheduler = 1`

2.2.8. Accesos a objetos de base de datos

Según Murach (2019) las bases de datos Mysql poseen diferentes formas de brindar accesos, tienen 4 niveles de privilegios.

1. Global: Refiere a todas las bases de datos
2. Base de datos: Hace referencia a una base específica.
3. Tabla: Asociado a todas las columnas de una tabla
4. Columna: Asociado a una columna específica.

2.2.8.1. Grant

Este comando es usado para brindar acceso a un determinado objeto de la base de datos sobre un nivel de privilegios se usa el comando Grant.

La sintaxis del comando Grant se muestra en la Figura 16.

Figura 16

Sintaxis del comando grant

```
GRANT
    priv_type [(column_list)]
    [, priv_type [(column_list)]] ...
ON [object_type] priv_level
TO user [auth_option] [, user [auth_option]] ...
[REQUIRE {NONE | tls_option [[AND] tls_option] ...}]
[WITH {GRANT OPTION | resource_option} ...]

GRANT PROXY ON user
    TO user [, user] ...
    [WITH GRANT OPTION]

object_type: {
    TABLE
  | FUNCTION
  | PROCEDURE
}

priv_level: {
    *
  | *.*
  | db_name.*
  | db_name.tbl_name
  | tbl_name
  | db_name.routine_name
}

user:
    (see Section 6.2.4, "Specifying Account Names")
```

```

auth_option: {
    IDENTIFIED BY [PASSWORD] 'auth_string'
    | IDENTIFIED WITH auth_plugin
    | IDENTIFIED WITH auth_plugin AS 'auth_string'
}

tls_option: {
    SSL
    | X509
    | CIPHER 'cipher'
    | ISSUER 'issuer'
    | SUBJECT 'subject'
}

resource_option: {
    | MAX_QUERIES_PER_HOUR count
    | MAX_UPDATES_PER_HOUR count
    | MAX_CONNECTIONS_PER_HOUR count
    | MAX_USER_CONNECTIONS count
}

```

Nota: <https://dev.mysql.com/doc/refman/5.6/en/grant.html>

En caso se requiera consultar los accesos de un usuario se usa el comando
SHOW

En la siguiente figura se muestra el uso del comando show

La sintaxis del comando **show** y un ejemplo se muestra en la Figura 17 y
Figura 18 respectivamente.

Figura 17

Sintaxis del comando Show

```
SHOW GRANTS [FOR user]
```

Nota: <https://dev.mysql.com/doc/refman/5.6/en/show-grants.html>

Figura 18

Un ejemplo para comando show

```
mysql> SHOW GRANTS FOR 'jeffrey'@'localhost';
```

```
+-----+
| Grants for jeffrey@localhost |
+-----+
| GRANT USAGE ON *.* TO `jeffrey`@`localhost` |
| GRANT SELECT, INSERT, UPDATE ON `db1`.* TO `jeffrey`@`localhost` |
+-----+
```

Nota: Obtenido desde: <https://dev.mysql.com/>

Se tienen las siguientes variantes de consulta de acceso en caso se desee consultar los accesos sobre el usuario como se muestra en la Figura 19.

Figura 19

Comando “show grants for”.

```
SHOW GRANTS;
SHOW GRANTS FOR CURRENT_USER;
SHOW GRANTS FOR CURRENT_USER();
```

Nota: <https://dev.mysql.com/doc/refman/5.6/en/show-grants.html>

2.2.8.2. Revoke

Según Murach (2019) después de haber creado y de haber brindado acceso a los usuarios, se tiene la necesidad de administrar estos privilegios, en caso sea necesario de retirar alguno de esos privilegios se usa el comando REVOKE.

La sintaxis del comando REVOKE se muestra en la Figura 20.

Figura 20.

Sintaxis del comando revoke.

```
REVOKE
    priv_type [(column_list)]
    [, priv_type [(column_list)]] ...
ON [object_type] priv_level
FROM user [, user] ...

REVOKE ALL [PRIVILEGES], GRANT OPTION
FROM user [, user] ...

REVOKE PROXY ON user
FROM user [, user] ...
```

Nota: <https://dev.mysql.com/doc/refman/5.6/en/revoke.html>

2.3. VIRTUAL PRIVATE NETWORK

Según Scott et al (1999) una VPN (Virtual Private Network) está compuesta por un grupo de computadores pertenecientes a una organización en donde los miembros de la misma comparten información y servicios. Un ejemplo de una Red privada es la LAN (Local Area Network) de la organización o también la WAN (Wide Area Network).

Se trata de una tecnología que simula una red privada dentro de una red pública, como Internet, utilizando conexiones virtuales temporales en lugar de enlaces físicos. Estas conexiones virtuales pueden establecerse entre dos dispositivos, entre un dispositivo y una red, o incluso entre dos redes distintas.

2.3.1. Intranet y VPN

Según Scott et al (1999) un usuario de una organización puede conectarse a su intranet de una organización si usa una VPN como medio de conexión, esta

conexión será segura. En el caso del supuesto que una organización deba brindar acceso a sus vendedores a un sistema web por donde puedan registrar sus ventas, en este escenario el sistema web solo se puede acceder desde dentro de la intranet de la empresa, para estos casos los usuarios(vendedores) de la organización deberán suscribirse a una VPN la cual les permitirá acceder a la intranet de la organización y mediante este acceso podrán usar el sistema web de ventas.

2.3.2. Riesgos

Scott et al. (1999) señalan que uno de los principales riesgos al utilizar una Red Privada Virtual (VPN) a través de internet es la posibilidad de que se intercepte información confidencial de los usuarios, por ejemplo, credenciales de acceso a los sistemas a los cuales ingresa, o también captura de la información sensible sobre clientes de la organización, direcciones de correo usados en las labores de día a día.

2.3.2.1. Firewall

Según Scott et al. (1999) un firewall o Cortafuegos sirve para lo mismo que un muro corta fuego es decir su función principal es que no se expanda el incendio, una de las funciones del firewall es no permitir paquete no autorizados de determinadas direcciones de internet y de determinados puertos preconfigurados puedan acceder a la red de la organización.

En caso que la organización quiera tener la opción que los empleados se conecten a la Red Interna desde fuera de la Red de la organización se debe realizar esta conexión a través de una VPN, la recomendación es que en caso

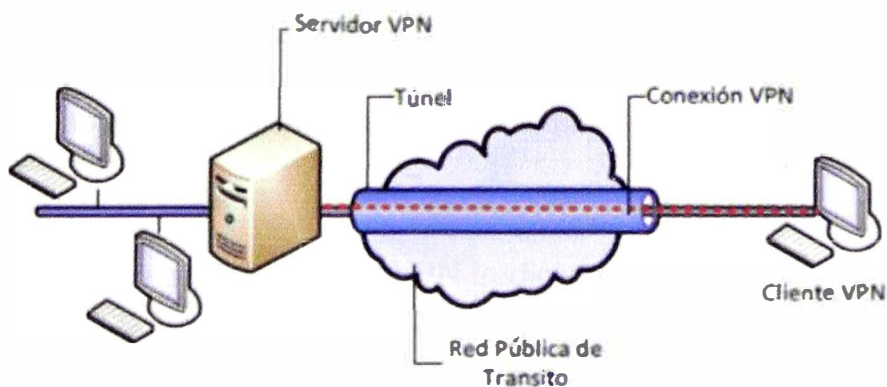
debe tener un Firewall, en caso la organización no posea un firewall la recomendación es que no se tenga la opción de VPN habilitada ya que será un gran problema de seguridad tener una VPN sin Firewall.

2.3.3. Componentes de una VPN

En su tesis, Rocha (2017) describe los elementos fundamentales de una VPN, los cuales incluyen un servidor VPN, un túnel para el tráfico de datos, la conexión VPN propiamente dicha, una red pública que sirve como medio de tránsito y el cliente VPN que se conecta al sistema. Estos componentes colaboran para proporcionar un entorno seguro para la transferencia de información a través de redes públicas, la conexión entre un cliente VPN y un servidor VPN se muestra en la Figura 21.

Figura 21

Conexión entre un cliente VPN y un servidor VPN.



Nota: CCNA Security Oficial Cert Guide

Los datos son encapsulados o empaquetados con una dirección de enrutamiento, con esta dirección los paquetes saben a qué destino llegar. Estos datos deben viajar cifrados para así viajar seguros por la red, la parte

en donde los datos son cifrados se conocen como “TUNEL”, en donde se realiza el cifrado y la encapsulación se conoce como conexión de la VPN.

2.3.4. Clasificación de VPN

Según Rocha (2017) tienen los siguientes tipos de VPN:

Una VPN de acceso remoto permite a los usuarios conectarse a la red interna de una organización a través de internet. Tras verificar sus credenciales, los usuarios pueden interactuar con la red de la misma forma que si estuvieran conectados físicamente a una red local, lo que facilita el acceso remoto a los recursos de la empresa, como si estuvieran presentes en la misma ubicación.

Una VPN punto a punto permite a los usuarios conectarse a un servidor VPN utilizando una conexión a internet. Una vez que se establece el túnel VPN, la comunicación entre los puntos se lleva a cabo de manera segura. Para que este tipo de conexión funcione correctamente, es esencial que la conexión a internet sea lo suficientemente estable y confiable, ya que de ello depende el rendimiento de la VPN.

Una VPN interna es similar a una VPN tradicional, pero la diferencia principal es que se accede a ella a través de la red local (LAN) de la sede en la que se encuentra el usuario. Este tipo de conexión ofrece un nivel de seguridad superior en comparación con una red Wi-Fi, ya que aprovecha la infraestructura de la LAN, que es menos vulnerable a las amenazas de redes inalámbricas.

Una VPN basada en firewall utiliza las capacidades de seguridad integradas en el servidor de firewall. Entre estas funcionalidades, se incluyen medidas para restringir el acceso a la red interna y la capacidad de realizar traducción

de direcciones IP. Estas características permiten controlar de manera efectiva quién puede acceder a los recursos internos de la red, ofreciendo así una capa adicional de protección.

En este tipo de soluciones el rendimiento del sistema operativo se puede ver afectado si no se tiene optimizado los recursos es decir se deben tener apagado los servicios innecesarios.

VPN basada en software:

Esta VPN es la idónea cuando ambos extremos pertenecen a diferentes organizaciones. Los productos software consumen el tráfico del túnel y este depende de la dirección y el protocolo, mientras que los productos basados en hardware encapsulan el tráfico y manejan el protocolo a manera independiente.

Algunas soluciones en software requieren cambios a nivel de las tablas de enrutamiento lo cual hace que el mantenimiento de estos sistemas con el tiempo sea más costoso.

2.4. JAVA

De acuerdo con Deitel & Deitel (2012), en 1991 Sun Microsystems inició un proyecto de investigación liderado por James Gosling, que dio lugar a la creación de Java, un lenguaje de programación orientado a objetos basado en C++. El propósito original de Java era desarrollar un lenguaje que permitiera escribir programas capaces de ejecutarse en diferentes tipos de

dispositivos, desde computadoras hasta otros tipos de aparatos. De ahí surgió la popular frase "write once, run anywhere", que significa que el programa se escribe una vez, pero puede ejecutarse en diferentes plataformas.

La popularidad de la web explotó en 1993, Sun vio el potencial de este lenguaje que podía generar contenido dinámico, actualmente se usa en mucho software empresarial, por ejemplo, en contenedores web, también es muy usado en los dispositivos smartphones, Smart-tv y una gran gama de dispositivos, multipropósitos.

En 2009 Sun Microsystems fue adquirido por Oracle, luego de esta adquisición Java ha sido el lenguaje de programación con la mayor cantidad de usuarios en el mundo.

2.4.1. JVM

Según Deitel & Deitel (2012) la JVM es la Java Virtual Machine de Java, ésta se encuentra del paquete JDK de java. Para ejecutar un programa en java, primero este se debe convertir en bytecodes, este es un archivo binario el cual tiene extensión "class", luego estos archivos serán ejecutados por la JVM, en caso se quiera ejecutar los bytecodes en diferentes plataformas es posible ya que JVM está disponible en una gran cantidad de dispositivos, tales como: smartphones, tablets, smartTVs, SmartWatch, autos, etc.

2.4.2. JDBC

Según Deitel & Deitel (2012) en java se tiene un package java.net el cual contiene todas las interfaces y classes JDBC, para usar este protocolo se

debe declarar en lo que se conoce como “cadena de conexión”, en esta cadena se encuentra el IP, nombre de la instancia de base de datos, puerto de conexión, esta cadena será utilizada por las clases de conexión. A través de este enlace se podrá enviar comandos y consultar información.

2.4.3. LOG4J

Es un paquete de código desarrollada por la Apache Software Foundation, el propósito es dejar constancia que cierta transacción se llevó a cabo en ejecución en un determinado tiempo de ejecución.

Los mensajes se pueden guardar en un archivo de texto en diferentes formatos (HTML, XML), los cuales poseen las siguientes etiquetas de acuerdo a sus niveles de prioridad:

FATAL, ERROR, INFO, WARN.

En las diferentes implementaciones de Log4j, posee un archivo de configuración, tiene por nombre: log4j.properties

2.4.4. Multiplataforma

Se mencionó en el punto 2.3.1. que Java puede ejecutarse en diferentes dispositivos, es decir los programas también se pueden ejecutar en diferentes versiones de Sistemas operativos, es decir un programa que se puede escribir y compilar en Windows. El archivo ejecutable (bitcode) del programa (.jar) se podrá ejecutar en los diferentes Sistemas Operativos que sean soportados por las versiones de JVM (Java Virtual Machine).

Según la información proporcionada por Java/Oracle en 2022, las ediciones 7.0 y 8.0 de Java son compatibles con una variedad de sistemas operativos. En el ámbito de Windows, esto incluye la versión 10 (desde la actualización 8u51), la serie 8.x (en su versión de escritorio), Windows 7 con Service Pack 1, Windows Vista con Service Pack 2, así como los servidores Windows 2008 R2 SP1, 2012 y 2012 R2 en su versión de 64 bits. Para los usuarios de Apple, se admite Mac OS X basado en procesadores Intel, específicamente las versiones 10.8.3 y superiores, así como 10.9 y posteriores. En cuanto a Linux, Java es compatible con múltiples distribuciones, entre ellas Oracle Linux (desde la versión 5.5 en adelante), Red Hat Enterprise Linux (también desde la 5.5 y versiones subsiguientes), Suse Linux Enterprise Server (a partir de la 10 SP2), y Ubuntu Linux (comenzando con la 12.04 LTS). Cabe mencionar que, dependiendo de la distribución específica y la versión de Java, podrían ser necesarias ciertas actualizaciones particulares para garantizar su funcionamiento óptimo.

2.4.7. Manejo de errores

El lenguaje de programación Java maneja excepciones, según Deitel & Deitel (2004), las excepciones/errores le permiten al programador manejar un código más limpio y más robusto y más tolerable a fallas. Los productos comerciales tienden a usar mucho más el manejo de errores, el uso del manejo de errores varía de la naturaleza del producto software.

El manejo de errores está diseñado para gestionar los errores síncronos, mas no está en su diseño poder resolver errores ante situaciones asíncronas, la

gestión de excepciones le permite al programador realizar alguna acción ante una situación en donde el programa ya no puede recuperarse, necesita realizar un proceso de salida ordenado y limpio, también poder ejecutar la salida del mismo dejando algún mensaje en los logs de la aplicación.

2.5. KANBAN

Según Ladas (2008) existen diferentes principios que se pueden aplicar para optimizar un proyecto de desarrollo de software, él considera 2 simples axiomas para poder alcanzar los objetivos en el proyecto a aplicar.

Axioma 1: Es posible dividir el proyecto en partes pequeñas que añadan valor y que puedan ser trabajadas de forma independiente.

Axioma 2: Es posible desarrollar el proyecto por partes las cuales incrementen valor durante un flujo de requerimientos de desarrollo.

2.5.1. Axioma 1

Según Ladas (2008) esta premisa no es solo aplicable al desarrollo de software sino también a todo desarrollo incremental e iterativo de algún producto, en el desarrollo de software Ágil se este concepto ha sido ampliamente estudiado por muchas décadas.

2.5.2. Axioma 2

Según Ladas (2008) este axioma es específico para la optimización de desarrollo de software, este axioma muestra de primera mano ciertos problemas en el desarrollo de software.

- Los requerimientos deben estar complemente definidos.
- El equipo de desarrollo debe ser tener características de un equipo de alto rendimiento.
- El flujo de trabajo y cronograma de trabajo debe estar ser óptimo y continuo quiere decir que las entregas deben realizarse de manera periódica con la menor cantidad de retrasos, retrabajos

2.6. LINUX

Según Siever et al. (2005) el sistema operativo Linux fue desarrollado por Linus Tovalds en la Universidad de Helsinki, este sistema contiene muchas aplicaciones para desarrolladores las cuales son muy confiables y con el paso de los años hay una gran comunidad que ha mejorado el núcleo de Linux en varios aspectos.

El sistema Linux desde sus inicios fue muy popular debido a que es muy robusto es decir no sufre caídas, al ser público su código kernel esto generó que los desarrolladores de la comunidad puedan generar aplicativos para diversas necesidades.

Linux puede ejecutar una gran cantidad de software, en la actualidad existen más de 200 distribuciones.

El sistema operativo Linux trae instalados el Java Runtime Enviroment (JRE) esta permite ejecutar scripts de java, los cuales sirven para ejecutar cierta lógica de actualización al cual depende del requerimiento solicitado.

2.6.1. Scripts

Según Siever et al. (2005) en las distribuciones Linux se tiene por efecto instalado el Java JRE este nos va a permitir ejecutar scripts.

En las organizaciones requieren constantemente determinadas necesidades asociadas a flujos de trabajo, por ejemplo necesitan ejecutar scripts de forma periódica y que estos actualicen cierta información en los sistemas, por ejemplo se requiere generar los recibos de pago para los clientes según un determinado ciclo de facturación, por ejemplo se requiere sincronizar 2 sistemas los cuales tienen problemas en la sincronización se ejecuta en horas de madrugada con la finalidad de que los sistemas cuenten con la misma información con un desviación máxima de 1 día. Unos de los scripts más importante en toda organización es la ejecución de carga la información asociada a la explotación de información o llamada Data Warehouse, les permitirá a los usuarios de toda la organización consultar la información de los sistemas sin consumir recursos en los sistemas legados.

2.7. NAVICAT

La herramienta software Navicat es un cliente el cual se conecta a diferentes repositorios de información de una forma amigable y sencilla, mostrando la información de las tablas muy práctica con muchas opciones visuales así dar una experiencia de usuario muy buena.

El uso de esta herramienta es con licencia ya que las funcionalidades que brinda le permiten al usuario leer y manipular la información de forma rápida y precisa.

Esta herramienta es muy versátil tiene diferentes opciones desde donde se pueden ejecutar queries, ver tablas y vistas, ejecutar procedures y funciones. Tiene una funcionalidad muy importante la cual le permite al usuario guardar los queries que usa y adicionalmente le permite guardarlos en carpetas, con esta opción los usuarios de la herramienta Navicat pueden guardar queries en donde requerías ejecutar reportes, o actualizar información y poder ejecutarlos desde el cliente de forma sencilla y amigable.

2.8. MARCO SCRUM

Desde la publicación de Kent et al. (2002) sobre el “Manifiesto of Agile Software Development” se indicaron 4 puntos importantes que sirvieron de base para su creación.

SCRUM es un marco de trabajo diseñado para aplicar metodologías ágiles en la creación, entrega y gestión de productos en diversas áreas. Su aplicación no se limita al desarrollo de software, ya que también es útil en marketing, la gestión de productos y otros sectores donde se requiera realizar iteraciones frecuentes y ofrecer resultados valiosos.

Según la Guía SCRUM Schwaber y Sutherland (2020), agrega valor a través de soluciones para problemas complejos.

Según Hron y Obwegeser (2022) el marco de trabajo SCRUM tiene artefactos, reuniones y roles los cuales siguen objetivos y siempre apuntan a lograr un producto funcional.

2.8.1. Pilares de SCRUM

Según Schwaber y Sutherland (2020) los pilares de SCRUM son:

- Transparencia: el trabajo realizado debe ser visible para todos los integrantes del equipo, en caso que el equipo no tenga acceso a visualizar todos los documentos, actas y coordinaciones formales que impliquen carga de trabajo, éste no tomará las mejores decisiones para el mejor desempeño del mismo, la transparencia es uno de pilares más importantes de SCRUM y que en él descansan los otros dos.
- Adaptación: Cuando se tiene un equipo autogestionado, es decir el equipo tiene autonomía en diferentes aspectos de la carga de trabajo, este puede adaptarse mucho mejor a los cambios durante el proyecto. Los cambios en un proyecto son una constante, durante la ejecución de un proyecto éste siempre tendrá algún cambio ya sea en documentación, procesos de entrega, pases a producción u otro aspecto asociado a algún proceso relacionado a la entrega de funcionalidades.
- Inspección: La inspección ayuda a detectar problemas que afecten la entrega de funcionalidades, en el marco SCRUM se indica que los eventos se deberían llevar a cabo con cierta frecuencia o tener una determinada cadencia, esta inspección o detección de problemas permite que la adaptación se pueda dar de forma natural ya que ambos valores son necesarios.

2.8.2. Valores de SCRUM

Según Schwaber y Sutherland (2020) nos indica que los valores son: compromiso, foco, franqueza, respeto y coraje.

De acuerdo con Wirdatul Hidayah et al. (2022), los principios de SCRUM están orientados a que el equipo de SCRUM alcance sus metas, resuelva los desafíos que se presenten y tome decisiones que contribuyan a su desarrollo. Esto implica que los miembros del equipo se vuelvan competentes, autónomos y que se fomente el respeto entre ellos.

2.8.3. Roles de SCRUM

Product Owner

Según Payano Muñoz (2019), el Product Owner es el encargado de mantener la comunicación entre los interesados y el equipo de desarrollo, además de establecer la prioridad de las historias de usuario durante la planificación, además es la persona que debe explicar al equipo de desarrollo las historias y en caso que el equipo de desarrollo tenga dudas a nivel funcional de alguna historia el Product Owner es el encargado de liberar de estas dudas al equipo de desarrollo.

El equipo de desarrollo

Según Kent et al, (2002) el equipo de desarrollo es el encargado de crear cualquier funcionalidad durante los Sprint, debe ser un equipo autoorganizado en donde cada integrante tiene habilidades específicas que son requeridas para crear las funcionalidades descritas en las historias de usuario.

Scrum Master

De acuerdo con Afshari y Gandomani (2022), el Scrum Master se encarga de asegurar que el equipo siga correctamente el marco de trabajo SCRUM. Su principal responsabilidad es guiar al equipo de desarrollo para que se concentre en la creación de las funcionalidades solicitadas en el backlog.

Entre sus funciones se incluyen:

- Ayuda con la liberación de impedimentos en el proceso del equipo de desarrollo.
- Asegurarse que se lleven a cabo todos los eventos SCRUM de formas positiva.
- Planificar y asesorar las implementaciones de SCRUM dentro de la organización.

2.8.4. Eventos de SCRUM

En la SCRUM Guide, Schwaber y Sutherland (2020) nos mencionan los siguientes eventos SCRUM:

Sprint: El núcleo de SCRUM es el sprint, el periodo en el que el equipo de desarrollo, junto con el Scrum Master, se dedica a crear las funcionalidades. Durante este tiempo, se deben tener en cuenta varios aspectos: no realizar modificaciones en las historias de usuario, no comprometer la calidad del producto y revisar el backlog con el Product Owner.

Cada Sprint se puede considerar como un mini proyecto y solo el Product Owner puede cancelar un Sprint esta acción solo se considera en caso las funcionalidades a desarrollar no agreguen valor al negocio.

Sprint Planning: Es una reunión en donde se establece el trabajo que se va a realizar en el Sprint, se abordan los siguientes temas:

- Comunica el valor del Sprint de los interesados al equipo de desarrollo, el porqué es importante el Sprint.
- Se comenta sobre las historias pendientes en el Sprint Backlog las cuales se requieren y cuales tienen más prioridad que otras.
- Se debe tratar de cómo se piensa lograr el trabajo, la forma a completar los objetivos del Sprint queda a criterio exclusivo de los desarrolladores.

Daily: Es una reunión de máximo 15 minutos en donde los desarrolladores lo usan para disminuir la complejidad, en caso que el Product Owner y el SCRUM Master deseen participar lo pueden hacer, pero es una reunión de los desarrolladores, se revisa impedimentos y se toma decisiones para poder levantarlos de ser el caso.

Sprint Review: Es una reunión en donde el SCRUM TEAM muestra el trabajo desarrollado durante el Sprint, en esta reunión se debe mostrar como el Product Backlog se puede ajustar a las nuevas necesidades del proyecto de los Stakeholders.

Sprint Retrospective: Esta reunión, dirigida por el Scrum Master, tiene como propósito planificar estrategias para mejorar la productividad, calidad y

efectividad del trabajo. Durante la sesión, se revisan los problemas que surgieron y las soluciones implementadas para resolverlos.

Artefactos de SCRUM:

Product Backlog: Es la lista de cómo mejorar el producto si las historias de usuarias tienen definición de terminado, son historias que podrían entrar al Sprint Planning en caso no lo tengan se debe realizar sesiones de refinamiento o tal vez es necesario que la funcionalidad se divida en donde más pequeñas.

Sprint Backlog: Los elementos a desarrollar durante el Sprint, junto con el plan para alcanzar el objetivo, conforman el sprint backlog. Este se va actualizando a lo largo del Sprint para monitorear el progreso de las historias de usuario y, a través de un tablero o panel informativo, identificar posibles problemas que puedan surgir en dichas historias.

Definición de terminado:

Es importante de cada historia de usuario esté definida explicando que es lo que se requiere y que se espera lograr con determinada funcionalidad, así como también se necesita tener los criterios de aceptación de todas las funcionalidades a trabajar por el equipo de desarrollo, esto le brinda un alcance específico de lo que busca tener en las historias de usuario para la organización.

2.9. DISEÑO DE BASE DE DATOS.

Según Ramakrishnan y Gehrke (2003) y Elmasri y Navathe (2007) el diseño de base de datos consiste en definir la estructura lógica y física de una o más bases de datos para responder a las necesidades de los usuarios con respecto a la información y para un conjunto concreto de aplicaciones.

El diseño de una base de datos es un proceso sistemático que estructura la información de un sistema. Este proceso involucra la definición detallada de elementos como tablas, relaciones, atributos y claves, asegurando que los datos se organicen de manera lógica y eficiente. Para lograr esto, es fundamental comprender las necesidades específicas de los usuarios y las aplicaciones que interactuarán con la base de datos.

El desarrollo de los sistemas de información contemporáneos demanda un acercamiento metodológico bien definido, dada su alta complejidad. La metodología basada en dividir el problema en partes más pequeñas ha probado ser altamente efectiva, permitiendo abordar el proceso mediante etapas ordenadas y asequibles. La evolución gradual del proyecto, comenzando con la identificación de necesidades hasta su implementación, facilita que cada paso se fundamente en los anteriores, resultando en un diseño consistente y eficaz.

La complejidad característica de los sistemas de información actuales hace necesaria una aproximación metodológica clara y definida para su desarrollo.

El método que consiste en segmentar el trabajo en unidades más reducidas ha demostrado su eficacia, facilitando la gestión del proceso mediante fases ordenadas y comprensibles. La implementación escalonada, desde la determinación de los requerimientos iniciales hasta la puesta en funcionamiento, permite que cada fase se apoye en los resultados previos, lo que conduce a un diseño íntegro y bien fundamentado.

A continuación, se determina el diseño lógico del repositorio de datos, optando por el esquema más conveniente (pudiendo ser relacional u orientado a objetos, entre otros). Este paso comprende la definición de los elementos principales, sus propiedades y los vínculos fundamentales que conformarán la base de datos, prescindiendo aún de las especificaciones técnicas de su implementación.

La siguiente fase consiste en elaborar el modelo físico, adecuando la estructura lógica para su implementación en un gestor de bases de datos particular. Este proceso abarca la mejora de los mecanismos de indexación, la especificación de las reglas de integridad y el ajuste de los elementos que determinan la eficiencia operativa.

Finalmente, se procede a la implementación y carga de datos, ajustando el diseño según sea necesario para garantizar un rendimiento óptimo.

Resulta fundamental señalar que el desarrollo de bases de datos no sigue un camino lineal, sino que es un proceso recurrente. Durante la evolución del proyecto, pueden emerger distintas necesidades o requisitos que hagan necesario reconsiderar y replantear decisiones anteriores. Asimismo, la arquitectura de la base de datos mantiene una vinculación directa con el diseño de los programas que la utilizarán, demandando una sincronización permanente entre los equipos responsables del desarrollo.

2.9.1. Fase de diseño de base de datos.

Fase 1. Levantamiento de información y análisis de requisitos

La etapa preliminar del desarrollo de una base de datos requiere un minucioso análisis de los requerimientos, expectativas y propósitos de los usuarios finales del sistema. Esta primera fase, denominada captación y estudio de requisitos, es crucial para lograr que la estructura de datos satisfaga plenamente las necesidades operativas de sus destinatarios.

Se divide en dos etapas se: la recopilación de información y luego el refinamiento de los requisitos.

La primera etapa en el diseño de una base de datos consiste en una exhaustiva investigación para identificar y comprender las necesidades de quienes utilizarán el sistema. Este proceso, conocido como recopilación de requisitos, implica un análisis detallado de las expectativas y demandas de los usuarios.

Es fundamental identificar a todos los actores involucrados, tanto usuarios directos como sistemas, y analizar cómo interactúan con la información. La información recopilada, a menudo dispersa e informal, debe ser organizada y estructurada.

Se deben considerar diversos aspectos, como: la estructura de los datos, los procesos que se realizarán sobre ellos, las restricciones de seguridad y rendimiento, y cualquier otro factor relevante para los usuarios. Para obtener esta información, se utilizan diversas técnicas como entrevistas, encuestas y análisis de la documentación existente.

La recopilación de requisitos es iterativa, lo que implica:

Identificar a los usuarios clave es determinar quiénes utilizarán la información y qué esperan obtener de ella.

Analizar la documentación existente: Revisar documentos, sistemas y procesos actuales.

Realizar entrevistas y encuestas: Obtener información directamente de los usuarios a través de diversas técnicas.

El objetivo principal de esta etapa es asegurar que la información estructurada resuelva las necesidades y expectativas de los usuarios, evitando así futuros problemas y rediseños.

Refinamiento de requisitos

Es crucial reconocer que las necesidades de los usuarios pueden evolucionar a lo largo del proyecto. Por ello, es fundamental mantener una comunicación constante con ellos para identificar y abordar cualquier cambio o inconsistencia en los requisitos. Involucrar activamente a los usuarios en el proceso de desarrollo no solo garantiza su satisfacción, sino que también reduce la probabilidad de errores en las etapas posteriores.

Una vez recopilados los requisitos, el siguiente paso es organizarlos y estructurarlos de manera formal. Esto se logra mediante técnicas de especificación, como la notación Z, que permiten representar los requisitos de forma clara y concisa utilizando diagramas, tablas y otros recursos visuales.

Esta etapa, aunque demandante en términos de tiempo y recursos, es esencial para el éxito del proyecto. Identificar y corregir errores en las primeras fases del desarrollo es mucho más eficiente y económico que hacerlo en etapas posteriores. Una correcta definición de los requisitos garantiza que la solución final satisfaga las necesidades del usuario y cumpla con los objetivos del proyecto.

En resumen, el refinamiento de los requisitos es un proceso iterativo que implica comunicación continua con los usuarios: Mantener una línea abierta de comunicación para identificar cambios y ajustes.

Identificación temprana de errores: Detectar y corregir cualquier inconsistencia o error en los requisitos para evitar problemas en etapas posteriores.

Fase 2. Diseño Conceptual

Según Elmasri y Navathe (2007), el diseño conceptual busca crear una representación abstracta y detallada del sistema, a partir de los requisitos recopilados en etapas previas. Esta fase se enfoca en construir un esquema conceptual que capture de manera precisa y comprensible las entidades, atributos y relaciones entre los datos, sin entrar en detalles técnicos sobre la implementación.

El objetivo principal del diseño conceptual es establecer una base sólida y unificada para el desarrollo del modelo ER. Este esquema conceptual será coherente con los requerimientos iniciales y servir como referencia para verificar que todas las necesidades del sistema estén contempladas.

En esta etapa, se prescinde de consideraciones tecnológicas específicas como el sistema de gestión de base de datos. El enfoque debe estar centrado en la estructura lógica de la información, dejando para etapas posteriores las decisiones relacionadas con la implementación.

El modelo ER

Existen diversas metodologías para representar de manera abstracta los requerimientos. Una ampliamente utilizada es el modelo entidad-relación. Este enfoque, por su sencillez y versatilidad, se ha convertido en una herramienta fundamental en el diseño conceptual.

Se centra en representar las entidades, atributos y relaciones que conforman el sistema a modelar. De esta manera, permite a los diseñadores construir una representación conceptual del mundo real que se desea. Su aplicabilidad trasciende las bases de datos, convirtiéndolo en una herramienta genérica y poderosa.

Es importante destacar que, aunque en español se encuentren ambas traducciones: modelo entidad-relación y modelo entidad-interrelación. Esta dualidad en la traducción se debe a las particularidades del idioma español al momento de traducir términos técnicos.

El lenguaje unificado de modelización

UML constituye un patrón industrial empleado para elaborar y registrar distintos aspectos de un sistema informático. Esta metodología incluye un conjunto extenso de representaciones gráficas que permiten examinar el sistema desde distintos enfoques. En el ámbito del diseño de almacenes de datos, destaca particularmente el diagrama de clases por su habilidad para esquematizar la disposición de la información. Estos diagramas plasman los elementos principales, sus propiedades y las interacciones entre estos, brindando una visión fija y abstracta del sistema de datos. La adaptabilidad y potencia expresiva de los diagramas de clases UML los ha transformado en un recurso indispensable para el diseño de estructuras de datos actuales.

Desarrollado y mantenido por la OMG, UML es un lenguaje de propósito general que se ha adoptado ampliamente en la industria del software. Su capacidad para modelar sistemas complejos lo convierten en una elección popular para modeladores y desarrolladores.

Fase 3. Diseño lógico

Previo al diseño pormenorizado del banco de datos, resulta crucial identificar la clase de sistema de almacenamiento que mejor satisfaga las exigencias del proyecto. Esta determinación, aunque no contempla aún la elección específica de un SGBD, establece las bases para el modelo lógico que se desarrollará posteriormente.

La estructuración lógica consiste en adaptar el modelo abstracto inicial, que no depende de ninguna plataforma particular, hacia un esquema lógico alineado con el tipo de almacén de datos escogido. En esta fase se adoptan las resoluciones técnicas pertinentes para ajustar el diseño conceptual a los requerimientos del gestor de bases de datos determinado. Ilustrando este punto, al seleccionar un sistema relacional, el esquema lógico resultante se compondrá de elementos como tablas, atributos y diversos tipos de claves.

Esta etapa se enfoca en los aspectos técnicos específicos del sistema de datos, considerando que la organización conceptual de la información ya fue determinada previamente. La meta es generar una estructura lógica minuciosa que servirá como fundamento para la implementación física del repositorio.

Este proceso puede dividirse en tres subetapas principales:

Revisión del modelo conceptual: Se realiza una revisión exhaustiva del modelo conceptual para identificar y corregir posibles errores o inconsistencias.

Adaptación del modelo abstracto: Se procede a convertir el diseño conceptual en un esquema lógico, adecuándolo a las particularidades del sistema gestor de almacenamiento seleccionado.

Optimización del esquema lógico: Se incorporan las reglas de normalización para preservar la uniformidad y consistencia de los datos en el banco de información.

Reconsideraciones del modelo conceptual

Se realiza un análisis del modelo conceptual desarrollado previamente, el objetivo de identificar y corregir posibles fallos que comúnmente surgen en este tipo de modelos. Es importante detectar y rectificar estos errores lo antes posible para evitar que se transmitan a las siguientes etapas.

Transformación del modelo conceptual en el modelo lógico

Este apartado abordará la organización lógica de sistemas de datos relacionales. Por tanto, la estructura lógica desarrollada será compatible con cualquier banco de datos de tipo relacional. Iniciaremos con los resultados obtenidos en la fase conceptual, representados a través de un diagrama de clases UML, y describiremos su proceso de conversión hacia un esquema basado en el modelo relacional. Este método ilustra cómo transformar los diversos elementos del diseño conceptual en componentes propios del modelo lógico, particularmente, del esquema relacional.

Normalización

Los fundamentos de la normalización se apoyan en principios matemáticos, incluyendo la teoría de conjuntos, la lógica y el álgebra relacional, para articular un conjunto de conceptos elementales que guían hacia una correcta organización de sistemas de almacenamiento relacionales.

Esta metodología implementa las formas normales (FN) como mecanismo para reconocer desviaciones de los principios óptimos de diseño. Un esquema relacional se considera conforme a una forma normal específica cuando satisface determinadas restricciones propias de ese nivel. La violación de estos criterios resulta en duplicidad de información y anomalías no deseadas en los procesos de actualización. Las FN se caracterizan por su naturaleza declarativa, estableciendo las condiciones necesarias sin detallar los procedimientos específicos para su implementación.

Fase 4. Diseño Físico

Es necesario seleccionar un Sistema de estructura de datos específico. Se debe analizar los diversos sistemas disponibles, tanto comerciales como de código abierto, y optar por un SGBD en el cual se pueda implementar el sistema de información desarrollado en las etapas previas del diseño.

Cada SGBD cuenta con componentes físicos propios. Los fabricantes emplean distintas tecnologías y enfoques para optimizar el rendimiento de sus sistemas. Previamente deberá ser ajustado considerando las particularidades. Se debe evaluar los aspectos de implementación y eficiencia.

El nivel físico y el nivel virtual

El estudio de las capas física y virtual de los repositorios de datos revela componentes como la disposición del almacenamiento y los mecanismos de acceso a los archivos del sistema. Cada gestor de bases de datos proporciona distintas alternativas para gestionar archivos y vías de acceso, siendo fundamental que el diseñador comprenda la implementación particular para maximizar la eficiencia del diseño físico.

Resulta esencial entender las particularidades de las operaciones que interactúan con el banco de datos, considerando aspectos como su periodicidad de ejecución y la magnitud prevista de información almacenada, para garantizar un óptimo desempeño del sistema.

Entre los aspectos fundamentales para determinar las configuraciones del diseño físico destacan:

Latencia de respuesta: Comprende el período transcurrido desde la emisión de una solicitud al gestor hasta la obtención de los datos. Una proporción considerable de este intervalo depende del sistema gestor, particularmente del tiempo requerido para acceder a la información solicitada. Existen factores externos, como la administración del sistema operativo o el acceso a dispositivos de almacenamiento, que escapan del control del gestor.

Optimización de espacio: Contempla el volumen de almacenamiento en disco consumido por los archivos del sistema y las estructuras de acceso, incluyendo índices y rutas adicionales.

Capacidad de procesamiento: Evalúa la cantidad media de operaciones gestionadas por unidad de tiempo, aspecto vital en sistemas que requieren procesamiento continuo de transacciones, como los utilizados en aviación o entidades financieras.

Transformación del modelo lógico en el modelo físico

Durante esta fase, se realiza la transformación del esquema lógico del repositorio de datos hacia un modelo físico, considerando el gestor de bases

de datos elegido para implementar el sistema, junto con las especificaciones del equipamiento, el sistema operativo y las aplicaciones base. Cada sistema gestor dispone de su lenguaje particular, creado por el fabricante, que facilita la implementación física del banco de datos, optimizándose para el entorno y maximizando el aprovechamiento del hardware, sistema operativo y el propio gestor. Frecuentemente, este lenguaje puede interpretarse como una ampliación del SQL estándar, incorporando elementos específicos que cada gestor requiere para establecer los componentes físicos. Sin embargo, existe una notable convergencia en muchos aspectos entre los diferentes gestores. El estándar SQL contempla la especificación de todos los elementos del diseño lógico, excluyendo los aspectos vinculados a la implementación física. La adaptación del esquema lógico a un gestor específico inicia con la definición de las tablas (incorporando atributos, claves principales, referencias y alternativas). Posteriormente, cada componente se asigna a un espacio apropiado en el nivel virtual y, finalmente, estos espacios se conectan con archivos físicos, correspondientes al nivel físico del sistema.

Fase 5. Implementación y Optimización

La etapa final del proyecto se centra en la implementación y optimización de la base de datos. En esta fase, se procede a cargar los datos iniciales, un paso crucial para evaluar el tamaño de las tablas y las características de las consultas más frecuentes. Posteriormente, se ajustará el modelo físico de la base de datos, lo que implica la configuración de índices y particiones para acelerar el acceso a los datos y mejorar significativamente el rendimiento del sistema. El objetivo principal es garantizar una respuesta rápida y eficiente a

las consultas de los usuarios, asegurando así una experiencia de usuario óptima

Procesamiento y optimización de consultas

Al revisar las consultas SQL generadas se busca revisar si los objetivos inicialmente planteados se están materializando. Por esta razón, es crucial comprender las operaciones que los SGBD relacionales llevan a cabo para ofrecer la respuesta esperada.

Los SGBD relacionales analizan distintas estrategias alternativas y eligen la más óptima. El procesamiento de consultas abarca todas las actividades que el SGBD realiza.

Este proceso se puede dividir en cuatro fases principales:

Descomposición de la consulta: consiste en convertir la consulta SQL en una representación interna basada en álgebra relacional.

Optimización de la consulta: implica seleccionar el plan de ejecución más eficiente entre diferentes alternativas. Este proceso se lleva a cabo en tres áreas: a) Optimización semántica, que se fundamenta en las restricciones definidas en el esquema de la base de datos. b) Optimización sintáctica, que transforma la expresión relacional original en una versión equivalente más eficiente. c) Optimización física, que selecciona el plan de evaluación con el menor coste, teniendo en cuenta los costos de cada operación según varios factores.

El proceso de optimización de consultas es fundamental para un SGBD relacional, ya que la estrategia seleccionada influye directamente en el rendimiento del sistema. La optimización es crucial en el diseño y construcción de un SGBD, ya que las técnicas aplicadas afectan el tiempo de resolución de consultas por parte del sistema, impactando su rendimiento global.

Procesamiento de vistas

Una vista es una tabla lógica que facilita el acceso a los datos de una o varias tablas a través de una consulta predefinida. No almacena información de

manera independiente, sino que se apoya en los datos de otras tablas. Las vistas proporcionan mecanismos de seguridad y ofrecen al diseñador de la base de datos la capacidad de controlar qué información está disponible para los usuarios.

La utilización adecuada de las vistas es crucial en el diseño eficiente de una base de datos, ya que permite ocultar la complejidad interna de las tablas y mantener la perspectiva del usuario, sin que este se vea afectado por los cambios en la estructura interna de las mismas.

Inicialmente, las vistas se utilizaban principalmente para simplificar las consultas, pero en la actualidad son esenciales en áreas como el diseño externo, los almacenes de datos y los sistemas distribuidos.

Administración de la seguridad

Finalmente, es crucial tener en cuenta las técnicas empleadas para proteger la base de datos contra accesos no autorizados, así como los mecanismos para gestionar la concesión y revocación de privilegios a los usuarios. La seguridad del SGBD es el componente responsable de estas y otras tareas, y su importancia ha crecido considerablemente, ya que la interconexión de ordenadores y dispositivos permite que cualquier persona pueda convertirse en usuario y posible atacante de una base de datos.

En muchas organizaciones, la información es vista como un activo intangible y sensible de gran valor. Para proteger este recurso, es indispensable asegurar el sistema de información y estar al tanto de las obligaciones legales que deben cumplirse.

CAPITULO III

DESARROLLO DEL TRABAJO DE INVESTIGACIÓN

3.1. ORGANIZACIÓN

Se mostrará el detalle de todas las actividades realizadas, desde las etapas de captura de requerimientos, planificación, desarrollo del proyecto.

En la realización del presente proyecto se usará el tablero Kanban para llevar un control sobre las actividades a realizar y así mantener el enfoque en cumplir el objetivo.

Se trabajó las siguientes historias de usuarios durante 11 Sprint.

1. Obtener la información en línea de la base de datos de CRM de los casos de los clientes.

Rol de Usuario: Encargado de campaña de casos

HU (Historias de usuarios): Se requiere que los casos al ser cargados se obtengan con un tiempo máximo de espera: 30 minutos desde la base de producción

CA (Criterio de aceptación): Que lo casos que se carguen en la tabla de MariaDB, como máximo debería tener fecha de creación de hace 30 minutos como máximo.

Detalle del script en Anexo 1.

2. Obtener la información del estado de BSCS de los clientes.

Rol de Usuario: Encargado de campaña de casos

HU (Historias de usuarios): Se requiere que se pueda tener información del estado del cliente en la plataforma BSCS (plataforma de facturación de la empresa)

CA (Criterio de aceptación): Contar con una columna para información del estado del cliente asociado a la plataforma BSCS.

Detalle del script en el Anexo 2.

3. Obtener la información en línea de la información de contacto de los clientes.

Rol de Usuario: Encargado de campaña de casos

HU (Historias de usuarios): Se requiere que la información de contacto de los clientes.

CA (Criterio de aceptación): Se debe mostrar teléfonos de contacto del cliente en caso que tenga más de un teléfono, cada teléfono se debe mostrar una columna diferente.

Detalle en el Anexo 3.

4. Obtener la información de las interacciones para poder validar si el proceso de videollamada generó ticket informativo sobre uso de app Claro Hogar.

Rol de Usuario: Encargado de campaña de casos

HU (Historias de usuarios): En el proceso de Videollamada se requiere saber si el asesor ingresó un id de interacción correcto el cual valida que el asesor informó de las características del App Claro Smarthome.

CA (Criterio de aceptación): El id de interacción ingresado por el asesor en caso que el id exista debe aparece en un campo adicional

5. Los casos deberán asignarse de forma automática a las diferentes contratas.

Rol de Usuario: Encargado de campaña de casos

HU (Historias de usuarios): En el proceso de atención de los tickets de casos, estos tickets al ingresar a la base de datos MariaDB (tabla: Casos ATF) se les debe asignar una contrata, considerar que se tienen 2 contratas.

CA (Criterio de aceptación): Los registros al ingresar a la tabla de Casos tendrá una columna en donde se distribuirá la cantidad de casos generados de acuerdo a un % por contrata. Por ejemplo, si se tienen 2 contratas MDY y Dynamicall, se debería tener un parámetro configurable asociado al porcentaje para Dynamicall o MDY.

6. Los casos deberán asignarse dentro de la contrata a los asesores activos según vista configurable de asesores.

Rol de Usuario: Encargado de campaña de casos

HU (Historias de usuarios): Los casos asignados a una determinada contrata deberán asignarse a un asesor de la misma contrata.

CA (Criterio de aceptación): Se deberán tener los siguientes criterios de aceptación.

- Los asesores se encontrarán en una lista configurable para cada contrata
- La asignación de los casos a los asesores debe seguir una distribución normal.
- Sólo se asignarán los casos a asesores activos.
- Tener una interfaz para la activación y desactivaciones de asesores.

Script en el anexo 4.

7. Historia de usuario columnas de información:

Rol de Usuario: Encargado de campaña de casos

HU (Historias de usuarios): Contar con los campos de información esencial para poder dar gestión a los casos.

CA (Criterio de aceptación): Tener los campos principales de gestión de información esencial para poder dar atención a los casos generados.

id caso: Es el código del caso generado por el CRM con el que se identifica a un caso.

ESTADO: Es el estado del caso generado en el CRM.

CUSTOMER_ID: Es el código del cliente en el sistema facturador. (BSCS)

FECHA_CREACION: Es la fecha de creación del caso

PROBLEMA: Es el primer nivel de detalle de tipificación de un caso al ser registrado

PRODUCTO: es la tecnología la cual pertenece el servicio del cliente, puede ser: DTH, HFC, LTE e INTERNET INALAMBRICO (IFI)

DNI: el DNI del cliente dueño del servicio

FIRST_NAME: Nombres del cliente dueño del servicio

LAST_NAME: Apellidos del cliente dueño del servicio

PLANO: Plano de la red fija donde se encuentra geográficamente el cliente

Hub: equipo de telecomunicaciones que brinda el servicio de internet y televisión a un grupo de planos.

Dirección: ubicación geográfica donde se encuentra instalado el servicio del cliente.

Canal: Grupo organizacional donde pertenece el asesor que ha generado el caso.

asesor creador: código de usuario del asesor que ha generado el caso

COLA: área unidad organizacional a donde se deriva un caso para su posterior atención, esto permite saber qué área va a revisar un determinado caso.

VARIACION: Es el segundo nivel de detalle de tipificación de un caso al ser registrado.

DEPARTAMENTO: Según la dirección de instalación del cliente mostrar en esta columna el nombre del departamento asociado a la dirección del servicio.

Detalle en el anexo 5.

8. Historia de usuarios columnas: caso_duplicidad, caso_duplicidad_hist

Rol de Usuario: Encargado de campaña de casos

HU: Se requiere contar con un campo que indique si el cliente posee un caso en ejecución.

CA: En el escenario que tenga un caso en ejecución este mostrará el mensaje:

"SI", en otro caso se mostrará mensaje en blanco.

9. Historia de usuario columnas: revision_dia, cuenta_con_smarthHome, se_instala_SmarthHome_en_llamada, motivo_no_instalacion, Ofrecimiento_de_herramienta, Aceptacion_cliente, Servicio_afectado_videollamada, Motivo_de_no_aceptacion, NUMERO, MARCA_MODELO, SISTEMA_OPERATIVO, OPERADOR,

Rol de Usuario: Encargado de campaña de casos

HU: Se necesita agregar los siguientes campos de gestión en donde los asesores pueden ingresar información del proceso de atención con videollamada.

CA: Para los campos solicitados el asesor podrá ingresar SI/No dependiendo sea el caso

10. Historia de usuario columnas: sot_asociado, Llamada_agendada

Rol de Usuario: Encargado de campaña de casos

HU: Se necesita en la atención del caso un campo donde el asesor pueda agregar una SOT, esta puede ser de postventa o un mantenimiento.

Se necesita un campo en donde el agente pueda registrar si hay alguna llamada agendada con el cliente asociado al caso.

CA: Al atender a un cliente, este puede necesitar que se le genere una sot de postventa o de mantenimiento. Tener un campo alfanumérico para poder ingresar SI o NO según el asesor agende la llamada con cliente asociado al caso.

11. Historia de usuario columnas: obs, modelo_equipo, MAC, Firmware

Rol de Usuario: Encargado de campaña de casos

HU: Se necesita un campo en donde el asesor que atiende el caso pueda indicar todo lo realizado en las atenciones con el caso, mediante este campo, se puede dar seguimiento entre asesores.

Se requiere un campo en donde durante la atención del caso para los escenarios de videollamada el cliente mencione el modelo de celular que posee.

Se requiere 2 campos en donde el asesor pueda registrar el número de serie Mac y el firmware asociado al equipo router del cliente y así poder usar esta información para establecer posibles fallas masivas en los equipos asociados al firmware del equipo

CA: Tener un campo de texto de longitud máxima 5000 caracteres.

Tener un campo selector de opciones en donde el asesor pueda elegir de acuerdo a una lista de modelos de equipos de celular, el modelo del celular del cliente.

Se requiere tener 2 campos en donde el asesor pueda modificar a demanda e ingresar valores alfanuméricos.

12. Historia de usuario columnas: estado_bscs

Rol de Usuario: Encargado de campaña de casos

HU: Se requiere un campo en donde se muestre el estado del contrato con esto lo que se espera lograr es que el asesor tenga una herramienta rápida

en donde vea si el cliente ha dado de baja el servicio el estado aparecerá "d" o si el cliente no ha pagado su recibo aparecerá "s" de suspendido.

CA: Mostrar el ultimo estado del contrato más reciente de un determinado customer_id

13. Historia de usuario columnas: id_inter_smarthome

Rol de Usuario: Encargado de campaña de casos

HU: Se requiere un campo en donde se pueda almacenar un id de 13 dígitos, es el identificador del CRM(TIMPROD), con este identificador se busca saber si el cliente realizó un reinicio a través del App Smarthome

CA: El nuevo campo debe permitir ingresar el código del CRM.

14. Historia de usuario columnas: Ticket_Remedy_movilidad

Rol de Usuario: Encargado de campaña de casos

HU: Se requiere un campo en donde se pueda almacenar un id de 13 dígitos, es el identificador del CRM(TIMPROD), con este identificador se busca saber si el cliente realizó un reinicio a través del App Smarthome

CA: El nuevo campo debe permitir ingresar el código del CRM.

15. Historia de usuario columnas: id_inter_smarthome

Rol de Usuario: Encargado de campaña de casos

HU: Se requiere un campo en donde se pueda almacenar un id de 13 dígitos, es el identificador del CRM(TIMPROD), con este identificador se busca saber si el cliente realizó un reinicio a través del App Smarthome

CA: El nuevo campo debe permitir ingresar el código del CRM.

16. Historia de usuario columnas: telf_servicio, num1, num2, num3, num4

Rol de Usuario: Encargado de campaña de casos, gerente del área

HU: Se requieren 5 campos en donde se muestren los diferentes teléfonos fijos para contactar a los clientes, estos números deberán ser obtenidos de las siguientes maneras:

1. Base del CRM (TIMPROD)
2. Números brindados en las llamadas cuando el cliente solicita un mantenimiento
3. Números brindados en las llamadas cuando el cliente se comunica con el call center.

17. Historia de usuario columnas: diagnostico, servicioafectado, inconveniente, estado_revision , condición.

Rol de Usuario: Encargado de campaña de casos, gerente del área

HU: Se requieren 5 campos que cumplan las siguientes reglas:

1. Diagnóstico: es un campo se debe calcular según la condición registrada por el asesor y según la tabla de valores relacionados.
2. servicioafectado e inconveniente: Ambos campos se deben hallar de acuerdo a la información a los sistemas legados de acuerdo al caso registrado.
3. estado_revision: es un campo se debe calcular según la condición registrada por el asesor y según la tabla de valores relacionados

4. condición: es un campo en donde el asesor registra al revisar el caso en gestión.

CA: Se debe contar con 5 campos en donde se validen las reglas mencionadas en la historia de usuario. Para el campo condición solo el asesor podrá registrar las opciones validadas en la tabla proporcionada por el equipo del Product Owner.

18. Historia de usuario columnas: codigo_asesor_asignado, prioridad

Rol de Usuario: Encargado de campaña de casos, gerente del área

HU: Se requiere 2 columnas con las siguientes características:

1. codigo_asesor_asignado: se mostrará el código del asesor el cual está revisando el caso en gestión.

2. prioridad: es una columna calculada en donde se mostrará 3 valores dependiendo la siguiente regla:

ALTO: el caso ha sido generado por el área de clientes premium de seguimiento de casos y cuenta con una antigüedad menor a 1 día de generado y el cliente no pertenece al segmento A.

MEDIO: el caso ha sido generado por el área de clientes premium de seguimiento de casos y cuenta con una antigüedad mayor a 1 día de generado y el cliente no pertenece al segmento A.

BAJO: el caso no ha sido generado por el área de premium de seguimiento de casos.

CA: El campo código_asesor deberá mostrarse el código del trabajador del asesor y el campo prioridad deberá ser dinámico y se mostrará según las restricciones establecidas en el apartado de historias de usuario.

19. Historia de usuario columnas: fecha_diagnostico, fecha_obs, fecha_recupera, fecha_cierre, usuario_cierre.

Rol de Usuario: Encargado de campaña de casos, gerente del área

HU: Se requiere 5 columnas con las siguientes características:

1. fecha_diagnostico: este campo se debe guardar la última fecha en la cual se actualiza el campo diagnóstico.
2. fecha_obs: este campo se debe guardar la última fecha en la cual se actualiza el campo obs
3. fecha_recupera: este campo mostrará última fecha de recuperación del caso.
4. fecha_cierre: este campo mostrará la fecha la cual se realizó el cierre del caso.
5. usuario_cierre: este campo mostrará el código de usuario el cual realizó el cierre del caso.

CA: El campo fecha diagnostico debe ser un campo tipo fecha y de cálculo automático según su regla de formación, de forma similar el campo fecha_obs, fecha_recupera, fecha_cierre.

El campo alfanumérico usuario de cierre y debe estar actualizado como máximo con un delay de 30 minutos.

20. [KPI_ATF] HU de columnas:

Rol de Usuario: Encargado de campaña de casos, gerente del área

HU: Se requieren las mismas columnas de la tabla ATF_Casos:

Id_caso, nombre_asesor_Asignado, fecha_creacion, usuario_generador, customer_id, DNI, estado, producto, problema, servicio_afectado, inconveniente, estado_revisión, condición, diagnostico, aplica_videollamada, ofrecimiento_herramienta, aceptación_cliente, motivo_de_no_aceptación, sot_asociado, fecha_recupera, fecha_obs, usuario_cierre,

CA: Los campos mencionados se copiarán de la tabla de casos de ATF con las mismas especificaciones de campos que la tabla ATF_Casos.

21. [KPI_ATF] HU de columnas:

Rol de Usuario: Encargado de campaña de casos, gerente del área

HU: se requieren las siguientes columnas se detallan las reglas de formación a continuación:

nombre_usuario_generador, operación_usuario_generador, área_usuario_generador, nombre_usuario_cierre, operación_usuario_cierre, área_usuario_cierre,

CA: Se requiere 6 columnas, los campos solicitados siguen las mismas reglas que la tabla Casos ATF, es decir cumplen con las siguientes características:

1. nombre_usuario_generador: es el nombre del usuario generador asociado al caso del reporte KPI_ATF.

2. operación_usuario_generador: es el nombre del área de operación del usuario generador asociado al caso del reporte KPI_ATF.

- reinci_caso_tec_7d: Muestra la cadena 'SI' en cuando el caso asociado al reporte KPI_ATF, tiene algún caso técnico reincidente durante los últimos 7 días.
- cat_fec_reinci_caso_tec: mostrar la categoría de fecha (1día, 2-3 días, 7 días, etc.) del caso técnico asignado a la mesa de 2do nivel asociados al cliente durante los últimos 7 días considerados antes de la fecha de generación del caso.
- id_caso_reinci_caso_tec_7d: mostrar código del último caso técnico asociados al cliente durante los últimos 7 días considerados antes de la fecha de generación del caso.
- serv_afec_reinci_caso_tec_7d: mostrar el “*servicio afectado*” del último caso técnico asociados al cliente durante los últimos 7 días considerados antes de la fecha de generación del caso.
- inconv_reinci_caso_tec_7d: mostrar el “*inconveniente*” del último caso técnico asociados al cliente durante los últimos 7 días considerados antes de la fecha de generación del caso.
- problema_reinci_caso_tec_7d: mostrar el “*problema*” del último caso técnico asociados al cliente durante los últimos 7 días considerados antes de la fecha de generación del caso.

23. [KPI_ATF] HU de columnas:

Rol de Usuario: Encargado de campaña de casos, gerente del área

HU: se requieren las siguientes columnas se detallan las reglas de formación a continuación:

sot_asociado_tipotrabajo, sot_asociado_codusu, sot_asociado_operacion,
sot_asociado_area_generadora, sot_asociado_min_motsol,
sot_asociado_grupo_motsol, FECHA_DILACION_ATENCION,
FECHA_DILACION_CIERRE, dilacion_atencion, dilacion_cierre.

CA: Las reglas a considerar para las siguientes columnas son las siguientes:

- sot_asociado_tipotrabajo: mostrar el *“tipo de trabajo”* de la sot asociada (la sot asociada es una sot generada por la mesa de 2do nivel, puede ser un mantenimiento o una postventa, dependiendo lo que necesite el cliente)
- sot_asociado_codusu: mostrar el *“codusu”* de la sot asociada.
- sot_asociado_operacion: mostrar *“la operación”* asociado al usuario generador de la sot asociada.
- sot_asociado_area_generadora: mostrar *“área generadora”* asociado al usuario generador de la sot asociada.
- sot_asociado_min_motsol: mostrar el *“mínimo motivo de solución”* asociado al usuario generador de la sot asociada.
- sot_asociado_grupo_motsol: mostrar *“grupo motivo solución”* asociado al usuario generador de la sot asociada.
- FECHA_DILACION_ATENCION: mostrar resta de días desde la fecha de atención hasta el día presente.
- FECHA_DILACION_CIERRE: mostrar resta de días desde la fecha de cierre hasta el día presente.

- `dilacion_atencion`: mostrar la categoría de fecha del campo `fecha_dilacion_atención`. Por ejemplo: 1 día, 2-3 días, 3-4 días más de 5 días, etc.
- `dilacion_cierre`: mostrar la categoría de fecha del campo `fecha_dilacion_cierre`. Por ejemplo: 1 día, 2-3 días, 3-4 días, más de 5 días, etc.

24. [KPI_ATF] HU de columnas:

Rol de Usuario: Encargado de campaña de casos, gerente del área

HU: se requieren las siguientes columnas se detallan las reglas de formación a continuación:

`techsee`, `duration`, `Cat_Duracion`, `problema_r`, `inconveniente_r`, `motivo_r`, `tipo`, `cat_inter_smarthome`, `id_inter_smarthome`

CA: Las reglas a considerar para las siguientes columnas son las siguientes:

- `techsee`: mostrar "si" cuando el caso aparezca en la base de `techsee` de llamadas.
- `duration`: mostrar la cantidad de tiempo de la última llamada del caso en la base de llamadas de `techsee` en el siguiente formato: "[minuto]:[segundo]"
- `Cat_Duracion`: mostrar la categoría de llamada del campo anterior, por ejemplo: <1 min, 2-3 min, 4-5 min, etc.
- `problema_r`: mostrar el problema asociado al caso reincidente del cliente durante los últimos 7 días.
- `inconveniente_r`: mostrar el inconveniente asociado al caso reincidente del cliente durante los últimos 7 días.

- motivo_r: mostrar el motivo asociado al caso reincidente del cliente durante los últimos 7 días.
- tipo: mostrar la categoría tipo de la sot asociada al caso.
- cat_inter_smarthome: mostrar la categoría de la última interacción tipo Smarthome asociada al cliente.
- id_inter_smarthome: mostrar el id_interacción de la última interacción tipo Smarthome asociada al cliente.

25. [KPI_ATF] HU de columnas:

Rol de Usuario: Encargado de campaña de casos, gerente del área

HU: se requieren las siguientes columnas se detallan las reglas de formación a continuación:

estado_revision_reincidente, condicion_reincidente, diagnostico_reincidente, reinc_caso_tec_7d_valido, serv_Afec_reinci_var_tec_7d, inconv_reinci_var_tec_7d, usuario_generador_reinci_var_tec_7d.

CA: Las reglas a considerar para las siguientes columnas son las siguientes:

- estado_revision_reincidente: mostrar el estado de la última sot reincidente a 7 días asociado al cliente.
- condicion_reincidente: mostrar la condición de la última sot reincidente a 7 días asociado al cliente.
- diagnostico_reincidente: mostrar el diagnóstico de la última sot reincidente a 7 días asociado al cliente.
- reinc_caso_tec_7d_valido: mostrar el código del caso atendido por la mesa 2do nivel

- serv_Afec_reinci_var_tec_7d: mostrar el campo servicio afectado de la última variación técnica de los últimos 7 días
- inconv_reinci_var_tec_7d: mostrar el campo inconveniente de la última variación técnica de los últimos 7 días
- usuario_generador_reinci_var_tec_7d: mostrar el código de usuario generador afectado de la última variación técnica de los últimos 7 días.

26. [KPI_ATF] HU de columnas:

Rol de Usuario: Encargado de campaña de casos, gerente del área

HU: se requieren las siguientes columnas se detallan las reglas de formación a continuación:

reinci_caso_tec_30d, dias_reinci_caso_tec_30d, reinci_var_tec_30d, dias_reinci_var_tec_30d.

CA: Las reglas a considerar para las siguientes columnas son las siguientes:

- reinci_caso_tec_30d: mostrar el código de caso del último caso técnico de los últimos 30 días
- dias_reinci_caso_tec_30d: mostrar la cantidad de días asociado al caso a la última variación técnica de los últimos 30 días.
- reinci_var_tec_30d: mostrar el código de caso de la última variación técnica en los últimos 30 días.
- dias_reinci_var_tec_30d: mostrar la cantidad de días del caso de la última variación técnica en los últimos 30 días.

27. [KPI_KPMG] HU de columnas:

Rol de Usuario: Encargado de campaña de casos, gerente del área

HU: se requieren las siguientes columnas se detallan las reglas de formación a continuación:

fec_gen_date, usuario_generador, nombre_usuario_generador, area_usuario_generador, ubicacion_usuario_generador.

CA: Las reglas a considerar para las siguientes columnas son las siguientes:

- fec_gen_date: Mostrar la fecha de generación de la sot asociada
- usuario_generador: Mostrar el código del usuario generador de la sot.
- nombre_usuario_generador: Mostrar el nombre del usuario generador de la sot.
- area_usuario_generador: Mostrar el área asociado al usuario generador de la sot.
- ubicacion_usuario_generador: Mostrar la ubicación asociado al usuario generador de la sot.

28. [KPI_KPMG] HU de columnas:

Rol de Usuario: Encargado de campaña de casos, gerente del área

HU: se requieren las siguientes columnas se detallan las reglas de formación a continuación:

customer_id, codsolot, tipotrabajo, motivo, distrito, provincia, departamento, fecha_cierre, usuario_cierre, nombre_usuario_cierre, area_usuario_cierre, ubicacion_usuario_cierre, area_cierre

CA: Las reglas a considerar para las siguientes columnas son las siguientes:

- customer_id: Mostrar el código del customer(cliente) asociado a la sot del presente reporte, las sot a considerar son todas las sot de mantenimiento, postventa de clientes con servicios fijos (HFC, FTTH).
- codsolot: Mostrar el código de la sot del presente reporte.
- tipotrabajo: Mostrar el tipo de trabajo asociado a la sot del presente reporte
- motivo: Mostrar el motivo asociado a la sot del presente reporte
- distrito: Mostrar el distrito asociado a la sot del presente reporte
- provincia: Mostrar la provincia asociado a la sot del presente reporte
- departamento: Mostrar el departamento asociado a la sot del presente reporte
- fecha_cierre: Mostrar la fecha de cierre asociado a la sot del presente reporte
- usuario_cierre: Mostrar el usuario de cierre asociado a la sot del presente reporte
- nombre_usuario_cierre: Mostrar el nombre de usuario asociado a la sot del presente reporte
- area_usuario_cierre: Mostrar el área de usuario asociado a la sot del presente reporte
- ubicacion_usuario_cierre: Mostrar la ubicación del usuario de cierre asociado a la sot del presente reporte
- area_cierre: Mostrar el de cierre asociado que cerró la sot del presente reporte

29. [KPI_KPMG] HU de columnas:

Rol de Usuario: Encargado de campaña de casos, gerente del área

HU: se requieren las siguientes columnas se detallan las reglas de formación a continuación:

min_motsol, min_grupomotsolucion, tipo, estado_revision, categorizacion_analisis, diagnostico

CA: Las reglas a considerar para las siguientes columnas son las siguientes:

- min_motsol: Mostrar el mínimo motivo solución asociado a la sot del presente reporte
- min_grupomotsolucion: Mostrar el grupo asociado al mínimo motivo solución asociado a la sot del presente reporte
- tipo: Mostrar el tipo o categoría asociada a la sot del presente reporte.
- estado_revisión: Mostrar el estado revisión asociado a la sot del presente reporte.
- categorizacion_analisis: Mostrar la categorización análisis de la sot del presente reporte.
- diagnostico: Mostrar el diagnóstico de la sot del presente reporte.

3.2. ACTIVIDADES REALIZADAS

3.2.1. FASE DE CONCEPCIÓN

En este capítulo presentaré las solicitudes iniciales necesarias antes de poder tener los aplicativos funcionando.

Se generaron las siguientes solicitudes:

1. Solicitud de servidor virtualizado con Sistema Operativo Linux.
2. Solicitud de acceso vía Firewall desde Dynamicall al servidor de Claro.

3. Solicitud de instalación de software gratuito Mysql.

3.2.2. FASE DE ELABORACIÓN

En la presente fase se mostrarán las actividades realizadas, dentro de las mismas se incluye las solicitudes administrativas durante la puesta en producción de los Sprint a realizar.

Se generaron las siguientes solicitudes a fin de poder tener la conectividad de la SEDE de Dynamicall/MDY con el servidor MariaDB:

- Solicitud de accesos al servidor Linux para administración y desarrollo de la base de datos Mysql.
- Solicitud de Firewall acceso a Sede Dynamicall a Servidor MariaDB.
- Solicitud de Firewall acceso a Sede MDY a Servidor MariaDB.

Las solicitudes antes mencionadas siguen un flujo de aprobación en donde dependiendo de la naturaleza de la solicitud puede ser aprobados por la jefatura o hasta la Subdirección.

Se muestra el cronograma de trabajo para los Sprint en la Tabla 1.

Tabla 1

Cronograma de Sprint

Sprint	Historias de usuario atendidas	tiempo
1	de HU1 al HU3	2 semanas de ejecución
2	de HU4 al HU 7	2 semanas de ejecución
3	de HU 8 al HU 12	2 semanas de ejecución
4	del HU 12 al HU 18	2 semanas de ejecución
5	del HU 19 al HU 24	2 semanas de ejecución

Nota: Elaboración propia.

Historias de usuario en donde brinda soporte a la campaña de Atención de Casos ATF: detallado en el punto anterior.

Reportes:

KPI_Analisis: detallado en el punto anterior

KPI_KPMG: detallado en el punto anterior

KPI_ATF: detallado en el punto anterior

Durante la elaboración del presente proyecto se desarrollaron los siguiente eventos SCRUM:

- Daily: Todos los días al inicio de labores se reunía el equipo de desarrollo para poder liberar algún impedimento que haya ocurrido.
- Sprint Review: El Product Owner y el equipo de desarrollo mostraban las funcionalidades a los Stakeholders del proyecto.
- Sprint Planning: Al inicio de cada Sprint se evaluaba que historias ingresarían al sprint se realizaba de forma conjunta entre el Product Owner y el equipo de desarrollo.

3.2.3. Fase de Construcción

En la fase de construcción se generaron las vistas a las tablas desde donde las contratistas: MDY, Dynamicall, y el personal interno de la gerencia accedían a los desarrollos.

Objetos generados:

1. Tabla de casos: ATC_Fija.Casos_ATF
2. View de accesos: viewCasosATF, viewCasosMDY, viewKPI_ATF, viewKPI_MDY, viewKPICasos_MDY, viewKPI_CasosATF
3. Tablas de auditoria: AuCasosATF, AuSegCasos
4. Reportes: KPI_KPMG, KPI_Analisis, KPI_ATF
5. Scripts en java de actualización

3.2.4. Fase de Entrega

Se realizó una serie de reuniones en donde se capacitó a las contratistas en el uso de la herramienta Navicat en donde se les enseñó a interactuar con las vistas y con las tablas.

Los puntos tratados en las reuniones fueron los siguientes:

- Configuración de la conexión a la base de datos.
- Uso de las interfaces: tablas, view y queries básicos.
- Herramientas visuales: filtros, actualización de datos.
- Exportación de datos a Excel y actualización masiva de información.

Durante la fase de entrega se generó un procedure en donde se visualiza los tiempos de ejecución de la reportería entregada, así como también se puede observar en caso de saturación logs sobre el proceso que la ocasiona.

3.2.5. Validaciones

Se realizaron 2 tipos de validaciones en el proyecto las validaciones a nivel técnico y a nivel funcional.

Las validaciones a nivel técnico fueron realizadas por el jefe de área en donde se revisó la arquitectura usada para el desarrollo de las historias de usuario.

Las validaciones a nivel funcionales se realizaron con el personal de América Móvil Perú S.A.C. a cargo de las campañas, dueños de los procesos y lo que se validó es que los campos muestren la información que deberían mostrar para todos los escenarios mencionados en el punto anterior.

CAPITULO IV

ANÁLISIS Y DISCUSIÓN DE RESULTADOS

Se han obtenido las siguientes mediciones antes de la implementación de la herramienta Navicat y después de la implementación para la campaña de Análisis de SOT:

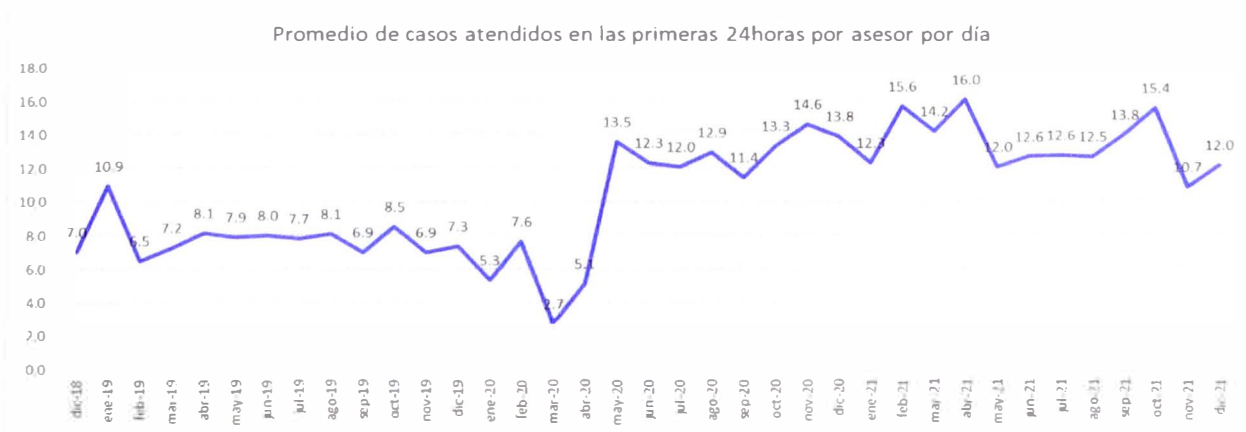
Variable: Atención al cliente, Dimensión: Eficiencia

Se realizó la medición de:

Cantidad de tickets promedios diarios atendidos dentro de las primeras 24 horas por asesor como se muestra en la Figura 22

Figura 22

Gráfico de tickets promedios diarios atendidos dentro de las primeras 24 horas por asesor

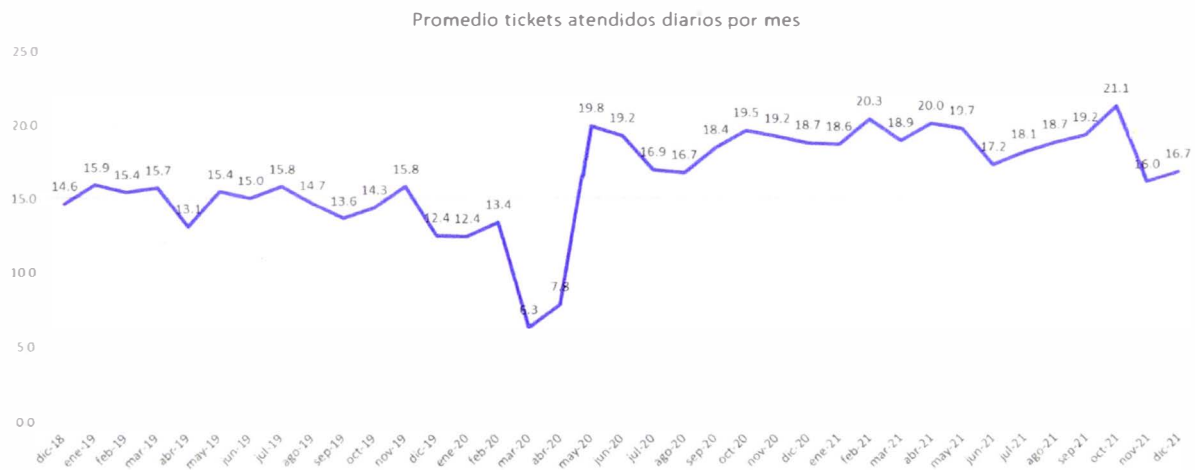


Nota: Elaboración propia

La Cantidad de tickets promedios diarios atendidos por asesor se muestra en la Figura 23.

Figura 23

Gráfico de tickets promedios diarios atendidos por asesor.



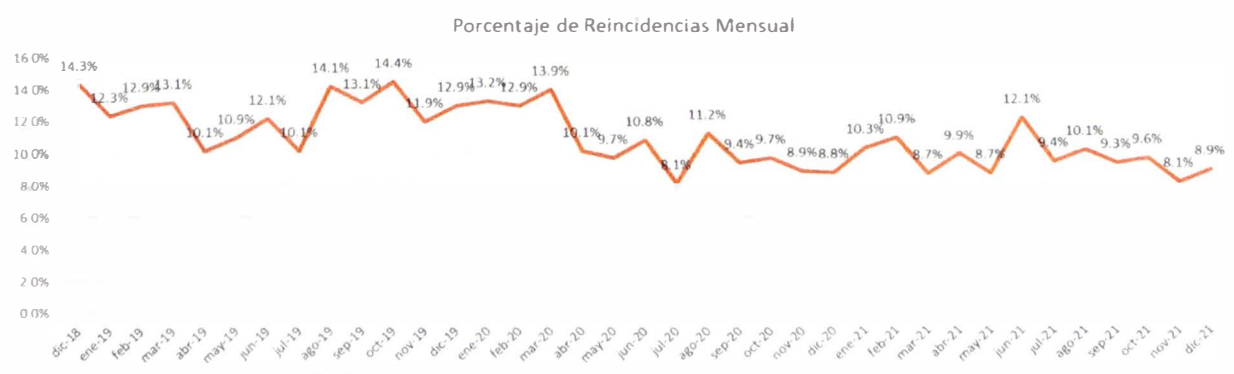
Nota: Elaboración propia

Dimensión: Eficacia:

La Tasa de Reincidencias por mes se muestra en la Figura 24.

Figura 24

Gráfico de la tasa de Reincidencias por mes



Nota: Elaboración propia

La Tasa de Solución Visita no efectiva por mes se muestra en la Figura 25.

Figura 25

Gráfico de la tasa de Solución Visita no efectiva por mes.



Nota: Elaboración propia

La implementación de la herramienta fue realizada en diciembre de 2019, durante los 3 meses siguientes se realizaron pruebas y se adaptó el Call center al nuevo flujo de trabajo.

Se ha comparado las métricas durante los últimos 6 meses antes de la implementación y los 6 primeros meses posterior a la implementación cuando el call center se adaptó a la herramienta.

Obteniendo los siguientes resultados:

1. En promedio los tickets atendidos diarios por asesor por mes antes de la implementación fueron de 13.7 tickets/día y luego de la implementación fue de 17.1 tickets/día con este resultado se verifica un incremento en la atención de tickets de un 71.4%.

2. En promedio la cantidad de atenciones realizadas por asesor por mes durante las primeras 24 horas después de generado los tickets antes de la implementación del sistema de soporte basado en base de datos fue de 7.6 tickets/día y luego de la implementación fue de 12.6 tickets/día con este resultado se verifica un incremento de atenciones durante las primeras 24 horas después de generado los tickets de 60%.

3. La métrica de reincidencias es muy usada en la operación de call center ya que es un indicador clave que indica que tanto se vuelve a comunicar un cliente que ya tenido atención del call center, se sabe que hay diferentes motivos por los cuales un cliente llama al área técnica y esto hace posible que las métricas mensuales sean medibles y comparables, el porcentaje de reincidencia del proceso de Análisis de SOT fue de 12.75% antes de la implementación y luego de la implementación fue de 9.81% con este cambio se logró un incremento del 42.85% con respecto a métrica posterior a la implementación del sistema de base de datos.

4. La métrica de visita no efectiva nos indica que tan efectivo fue el call center con sus procesos de descartes ya que usa lo indicado en el motivo de resolución de ticket del área de operaciones para poder verificar la calidad en sus procesos de descartes, el porcentaje de visitas no efectivas fue de 8.5% antes de la implementación y luego de la implementación bajó a 5.2% con este cambio se logró una mejora del 63.4% con respecto a la implementación del sistema de interfaces.

CONCLUSIONES

1. Solución a medida presenta mucha versatilidad antes una solución propietaria.
2. El costo de realizar los cambios a un sistema desarrollo in-house(interno) es mucho menor a un sistema propietario.
3. Los costos de tener un sistema desarrollo in-house(interno) son menores vs los altísimos costos de un comprar un sistema propietario.
4. Las caídas del sistema van a recaer en personal interno de la empresa, mientras que en un sistema propietario se revisará el acuerdo SLA.
5. Las limitaciones de tener un sistema de base de datos no permitirá mostrar alertas tipo pop-up al no tener un interfaz web o de otro tipo.
6. Los cambios y desarrollos deben estar acompañados de su respectiva documentación ya que en el caso que los mantenimientos de los sistemas in-house(interno) los realice una empresa contratista, éste debe contar con todas las herramientas para poder realizar la documentación solicitada.

RECOMENDACIONES

1. Se recomienda al jefe del área de TI realizar las modificaciones en los procesos a fin de que los aplicativos que tengan un scope de gerencia puedan estar dentro del ecosistema de aplicativos de TI.
2. Se recomienda al jefe de área operativa pueda usar la información obtenida de los sistemas en cuadros de mando en un aplicativo de inteligencia de negocio como por ejemplo Power BI.
3. Se recomienda al equipo de desarrollo, que ejecutó las historias de usuario realicen un aplicativo en donde consideren un dominio cloud, es decir tengan alojadas las diferentes partes de aplicativo en la nube de Azure o de Amazon (AWS) por ejemplo.
4. Se recomienda tener replicación en la base de datos ya que, al tener data sensible, la operación se puede ver afectada en caso se dé un evento de caída de servidor de base de datos.

REFERENCIAS BIBLIOGRÁFICAS

- Adiego Rodríguez, J., & Llanos Ferraris, D. (2010). *Fundamentos de informática y programación en C*. Paraninfo.
- Afshari, M., & Gandomani, T. (2022). A novel risk management model in the Scrum and extreme programming hybrid methodology. *International Journal of Electrical and Computer Engineering*, 12(3), 2911~2921.
<https://doi.org/http://doi.org/10.11591/ijece.v12i3.pp2911-2921>
- Arenal Laza, C. (2019). *Gestión de la atención al cliente / consumidor*. UF0036. TUTOR FORMACIÓN.
- De La Cruz Fernández, R. N. (2017). Eficacia y atención a los clientes en la empresa constructora TS&C, Huaraz 2017. Huaraz.
<http://repositorio.usanpedro.edu.pe/handle/USANPEDRO/9603>
- Deitel, P., & Deitel, H. (2004). *Cómo programar en C++ y Java*.
- Deitel, P., & Deitel, H. (2012). *Java How to Program*. Prentice Hall.
- Galitz, W. O. (2007). *The Essential Guide to User Interface Design*. Wiley.
- García Toribio, G., Polvo Saldaña, Y., Hernández Mora, J. J., Sánchez Hernández, M. J., Nava Bautista, H., Collazos, C. A., & Hurtado Alegría, J. A. (2019). Medición de la usabilidad del diseño de interfaz de usuario con el método de evaluación heurística: dos casos de estudio. *Revista*

Colombiana de computación, 20(1), 23-40.

<https://doi.org/http://dx.doi.org/10.29375/25392115.3605>

Gil López, J. V. (2019). *Fundamentos de atención al cliente*. España: E-learning S.L.

Hron, M., & Obwegeser, N. (2022). Why and how is Scrum being adapted in practice: A systematic review. *Journal of Systems and Software*, 183.

<https://doi.org/https://doi.org/10.1016/j.jss.2021.111110>

Java/Oracle. (10 de marzo de 2022).

<https://java.com/es/download/help/sysreq.html>

Jiménez Paneque, R. (2004). Indicadores de calidad y eficiencia de los servicios hospitalarios: Una mirada actual. *Rev. Cubana Salud Pública*, 30(1). Retrieved 26 de 11 de 2023, from http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-34662004000100004&lng=es&tlng=es

Kadenic, M., Koumaditis, K., & Junker-Jensen, L. (2023). Mastering scrum with a focus on team maturity and key components of scrum. *Information and Software Tecnology* (153).

Kent, B., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., & Fowler, M. (2002). Manifiesto Agile. <https://agilemanifesto.org/>.

Ladas, C. (2008). *Scrumban and other essays on KanBan Systems for Lean Software Development*.

Lima Rocha, L. E. (2017). *Implementación de una Red Privada Virtual Dinámica multipunto DMVPN con protocolos IPSEC, MGRE y NHRP*.

- Miranda Torres, R. J., & Blax Arteaga, N. G. (2018). Sistema Informático basado en plataforma Web para mejorar el proceso de gestión documental en una facultad de la universidad nacional de uayali. Pucallpa, Perú.
- Mora Rioja, A. (2014). *Bases de datos. Diseño y gestión*. Editorial Sintesis.
- Murach, J. (2019). *Murach's Mysql*. Mike Murach & Associates, Inc.
- Muther, A. (2002). *Customer Relationship Management*. Springer.
<https://doi.org/https://doi.org/10.1007/978-3-642-56222-8>
- Osorio Rivera, F. L. (2008). *Base de datos Relacionales*. Medellín, Colombia: Instituto Tecnológico Metropolitano.
- Payano Muñoz, D. (2019). Implementación de la metodología SCRUM para agilizar proceso de atención de requerimientos del departamento de tecnología de la información de la Caja Huancayo.
- Piñeiro Gómez, J. M. (2014). *Diseño de bases de datos relacionales*. Paraninfo.
- Pisco Gómez, A., Regalado Jalca, J. J., Gutiérrez García, J., Quimis Sánchez, O., Marcillo Parrales, K., & Marcillo Merino, J. (2017). *Fundamentos sobre la gestión de base de datos*. Alicante: Área de Innovación y Desarrollo S.L.
- Schwaber, K., & Sutherland, J. (2020). La Guía de SCRUM.
- Scott, C., Wolfe, P., & Erwin, M. (1999). *Virtual Private Networks*. O'Reilly.
- Siever, E., Weber, A., Figgins, S., Love, R., & Robbins, A. (2005). Linux in a Nutshell a desktop quick reference.

Wirdatul Hidayah, N., Rahayu Sasmita, R., Kurnia Mayangsari, M., Wira Kusuma, O., Rante, H., & Fariza, A. (2022). Invitin Project: Scrum Framework Implementation in a Software Development Project Management. *INTEK Jurnal Penelitian*, 9(1), 58-65.
<https://doi.org/http://dx.doi.org/10.31963/intek.v9i1.3332>

ANEXOS

ANEXO 1:

```
1 package prev;
2
3 import java.sql.Connection;
4
13 public class Variacion {
14
15     static Logger log = Logger.getLogger(Variacion.class.getName());
16
17     public static void main(String[] args) {
18         log.info(" ##--## Inicio de programa Descarga Global de Variaciones ##--## ");
19         log.info("Conectado a Base Oracle TIMPROD12C");
20         int ult_hora = 0;
21         if( args.length > 1 ){
22             Calendar cal = Calendar.getInstance();
23             Calendar cal2 = Calendar.getInstance();
24             int hora = 0, hora2 = 0, cont = 0, i = 0, j = 0, min = 0, aux = 0;
25             int day_ini = Integer.valueOf(args[0]);
26             int day_fin = Integer.valueOf(args[1]);
27             try{
28                 log.info("Statement creada para consultar la DB TIMPROD .... esperando respuesta...");
29                 while(true){
30                     cal = Calendar.getInstance();
31                     hora = cal.get(Calendar.HOUR_OF_DAY);
32                     min = cal.get(Calendar.MINUTE);
33                     log.info("Hora obtenida del servidor : "+hora);
34                     if( hora > 5 ){
35                         cont = 0;
36                         log.info("## -- Inicio de programa Descarga de Variaciones -- ##");
37                         Actualizacion_Variacion(day_fin,day_ini);
38                         log.info(" ## -- FIN Descarga de Variaciones -- ## ");
39                         //log.info(" Antes de validacion : ult_hora = "+ult_hora+ " , hora = " + hora);
40                         log.info("## -- Inicio de programa Cambio de estado en los casos --##");
41                         Actualizacion_Estado();
42                         log.info(" ## -- FIN del Cambio de estado en los casos -- ## ");
43                         if( aux==1 ) {
44                             log.info("## -- Inicio de programa Cambio de estado en los casos --##");
45                             Actualizacion_Estado();
46                             log.info(" ## -- FIN del Cambio de estado en los casos -- ## ");
47                         }
48                     }
49                 }
50             } catch (Exception e) {
51                 log.error("Error al ejecutar la consulta: "+e.getMessage());
52             }
53         }
54     }
55 }
```

```

49         Thread.sleep(30000);
50         //Actualizacion_VariacionUpd();
51     }
52     else{
53         if (cont == 0){
54             i=1;
55             j=2;
56             cont = 1 ;
57         }else{
58             log.info("Limites de consulta [ min=-"+j*2+" ,max=-"+i*2+" ]");
59             Actualizacion_Variacion(j*2,i*2);
60             i++;
61             j++;
62         }
63     }
64     Thread.sleep(30000);
65     aux = 0;
66     cal2 = Calendar.getInstance();
67     hora2 = cal2.get(Calendar.HOUR_OF_DAY);
68     if(ult_hora != hora2)
69         aux = 1;
70     ult_hora = hora2;
71     log.info(" Validacion : ult_hora = "+ult_hora+ " , hora2 = " + hora2);
72 }
73 }catch(Exception e){
74     e.printStackTrace();
75 }
76 }
77 }
78 }
79 }
80 private static void Actualizacion_Variacion(int min, int max) {
81     Connection conn_mysql = null, conn_mysql2 = null, conn_oracle = null;
82     Statement stmt_mysql = null, stmt_mysql2 = null, stmt_oracle = null ;
83     ResultSet rs_oracle = null;
84     String sql_mysql = "", sql_mysql2 = "", id_caso = "", nombre = "",
85         id_interaccion="", fecha_creacion="", apellido = "", phone="", act_code="",
86         x_customer_id = "";
87     int i = 0 ;

88     try{
89         conn_oracle = OracleConnTIPPROD.conectar();
90         stmt_oracle = conn_oracle.createStatement();
91         conn_mysql = ConnMYSQL.conectar();
92         conn_mysql2 = ConnMYSQL.conectar();
93         log.info("Conectado a Base MariaDB");
94         stmt_mysql = conn_mysql.createStatement();
95         stmt_mysql2 = conn_mysql2.createStatement();
96         log.info("Ejecutando query en la TIPPROD12C, extrayendo tipificaciones .... ");
97         rs_oracle = stmt_oracle.executeQuery("select ca.id_numero id_caso, "+
98             "i.interact_id, "+
99             "i.create_date, "+
100             "c.x_doc_num dni, "+
101             "i.phone customer_id, "+
102             "replace(replace(replace(replace(i.first_name,chr(39)),chr(45)),chr(92)),chr(52)), "+
103             "replace(replace(replace(replace(i.last_name,chr(39)),chr(45)),chr(92)),chr(52)), "+
104             "i.reason_1 producto, "+
105             "i.reason_2 variacion, "+
106             "i.reason_3 problema, "+
107             "b.s_title estado, "+
108             "det.servafect, "+
109             "det.inconven, "+
110             "e.s_title Cola, "+
111             "d.login_name asesor, "+
112             "j.act_code, "+
113             "j.entry_time, "+
114             "replace(replace(replace(replace(j.addnl_info,chr(39)),chr(45)),chr(92)),chr(52)) , "+
115             "c.x_customer_id "+
116             "from table_interact i "+
117             "left join table_case ca on i.x_interact2case = ca.objid "+
118             "left join table_contact c on c.objid = i.interact2contact "+
119             "left join sa.table_interact_subclasedet det on det.interact_id = i.interact_id "+
120             "left join sa.table_gbst_elm b on b.objid = ca.casests2gbst_elm "+
121             "left join sa.table_queue e on e.case_currq2queue = e.objid "+
122             "left join sa.table_user d on i.agent = d.login_name "+
123             "left join sa.TABLE_ACT_ENTRY j on ca.objid = j.act_entry2case "+
124             "where i.reason_1 in ('HFC','LTE','OTH POST','INTERNET INALAMBIRICO','FTTH') "+
125             "and i.create_date >= sysdate -"+min+" -0.1 "+
126             "and i.create_date <= sysdate -"+max+" + 0.1 ");

```

```

127
128      /*
129      rs_oracle = stmt_oracle.executeQuery("select a.id_number id_caso, "+
130      "a.creation_time fecha_creacion, "+
131      "c2.x_doc_num dni, "+
132      "c2.x_customer_id customer_id, "+
133      "c2.first_name, "+
134      "c2.last_name, "+
135      "c2.phone, "+
136      "e.s_title Cola, "+
137      "a.s_title tipificación, "+
138      "a.s_x_reason_1 producto, "+
139      "a.s_x_reason_2 variacion, "+
140      "a.s_x_reason_3 Problema, "+
141      "a.s_x_reason_4, "+
142      "b.s_title estado, "+
143      "a.x_case2curr_result , "+
144      "d.login_name asesor "+
145      "from sa.table_case a, "+
146      "sa.table_gbst_elm b, "+
147      "sa.table_user d, "+
148      "sa.table_queue e, "+
149      "sa.table_contact c2 "+
150      "where a.casests2gbst_elm = b.objid "+
151      "and a.case_owner2user = d.objid "+
152      "and c2.objid = a.case_reporter2contact "+
153      "and a.case_currq2queue = e.objid(+) "+
154      "and a.creation_time >= sysdate-3 "+
155      "and a.s_x_reason_1 = 'HFC' ");
156      */
157      Log.info("Consulta recuperada de TIMPROD");
158      i=0;
159      stmt_mysql2.executeUpdate("truncate Prueba.Variacion_temp");
160      Log.info("Tabla truncada: Variaciones_temp ");
161      Log.info("Insertando registros a MariaDB, tabla: Variaciones_temp ...");
162      while(rs_oracle.next()){
163          id_caso = rs_oracle.getString(1);
164          id_interaccion = rs_oracle.getString(2);
165          phone = rs_oracle.getString(5);

```

```

166     nombre = rs_oracle.getString(6);
167     apellido = rs_oracle.getString(7);
168     fecha_creacion = rs_oracle.getString(3);
169     act_code = rs_oracle.getString(16);
170     x_customer_id = rs_oracle.getString(19);
171     sql_mysql = "insert into Prueba.Variacion_temp( "
172         + "ID_CASO,id_interaccion,FECHA_CREACION,DNI,phone,FIRST_NAME, " // 1-6 columnas
173         + "LAST_NAME,PRODUCTO,VARIACION,PROBLEMA, ESTADO, " // 7-11 col
174         + "servicioafectado,inconveniente,cola,asesor, act_code , entry_time, "
175         + "addnl_info, x_customer_id ) values( " // 12-14 col
176         + id_caso + "," + id_interaccion + ","
177         + GetNull(fecha_creacion) + "," + rs_oracle.getString(4) + ","
178         + GetNull(phone) + ","
179         + GetNull(nombre) + ","
180         + GetNull(apellido) + ","
181         + GetNull(rs_oracle.getString(8)) + ","
182         + GetNull(rs_oracle.getString(9)) + ","
183         + GetNull(rs_oracle.getString(10)) + ","
184         + GetNull(rs_oracle.getString(11)) + ","
185         + GetNull(rs_oracle.getString(12)) + ","
186         + GetNull(rs_oracle.getString(13)) + ","
187         + GetNull(rs_oracle.getString(14)) + ","
188         + GetNull(rs_oracle.getString(15)) + ","
189         + act_code + ","
190         + GetNull(rs_oracle.getString(17)) + ","
191         + GetNull(rs_oracle.getString(18)) + ","
192         + x_customer_id + " ) ";
193     stmt_mysql.executeUpdate(sql_mysql);
194     //log.info(sql_mysql);
195     i++;
196 }
197 log.info("En tabla Prueba.Variacion_temp se insertaron "+ i + " registros");
198 log.info("Ejecución del procedure: Prueba.Variacion_temp en curso...");
199 stmt_mysql2.executeUpdate("call Prueba.Variacion_temp()");
200 }catch(Exception e){
201     e.printStackTrace();
202     log.error("Mensaje de la excepcion : "+e.getMessage());
203     log.error("ID_interaccion = "+id_interaccion);
204     log.error("Insert ejecutado en mysql = "+sql_mysql);

```

```

205         log.error("Select de validacion sobre id_interaccion "+sql_mysql2);
206         log.error("id_caso="+id_caso);
207         log.error("id_interaccion="+id_interaccion);
208         log.error("fecha_creacion="+fecha_creacion);
209         log.error("nombre="+nombre);
210         log.error("apellido="+apellido);
211         log.error("phone="+phone);
212         StackTraceElement[] stack = e.getStackTrace();
213         log.error("Error Class: "+e.getClass());
214         for( StackTraceElement ste : stack ){
215             log.error( ste );
216         }
217     }finally {
218         if (rs_oracle != null)
219             try {
220                 rs_oracle.close();
221             } catch (Exception e) {
222             }
223         if (stmt_oracle != null)
224             try {
225                 stmt_oracle.close();
226             } catch (Exception e) {
227             }
228         if (conn_mysql != null)
229             try {
230                 conn_mysql.close();
231             } catch (Exception e) {
232             }
233     }
234 }
235
236 private static String GetNull(String campo) {
237     if( campo == null )
238         campo = "NULL";
239     else
240         campo= ""+ campo.replace("'", "") +"";
241     return campo;
242 }
243
244 private static void Actualizacion_Estado() {
245     Connection conn_mysql = null, conn_mysql2 = null , conn_oracle = null ;
246     Statement stmt_mysql = null, stmt_mysql2 = null , stmt_oracle = null ;
247     ResultSet rs_mysql = null, rs_oracle = null;
248     int i = 0;
249     String id_caso = "",
250           sql_oracle = "", estado = "", id_caso_zero = "";
251     try{
252         conn_oracle = OracleConnTIPROD.conectar();
253         stmt_oracle = conn_oracle.createStatement();
254         //log.info("Conectado a Base Oracle");
255         conn_mysql = ConnMYSQL.conectar();
256         conn_mysql2 = ConnMYSQL.conectar();
257         log.info("Conectado a Base MariaDB");
258         stmt_mysql = conn_mysql.createStatement();
259         stmt_mysql2 = conn_mysql2.createStatement();
260         /*
261         rs_mysql = stmt_mysql.executeQuery("select id_caso from Prueba.Variacion "
262             + "where id_caso is not null "
263             + "and estado not in ('CERRADO','CONCLUIDO','SOLUCIONADO') ");
264         */
265         rs_mysql = stmt_mysql.executeQuery("select id_caso from ATC_Fija.CasosATF "
266             + "where id_caso is not null "
267             + "and estado not in ('CERRADO','CONCLUIDO','SOLUCIONADO') ");
268         while(rs_mysql.next()){
269             id_caso = rs_mysql.getString(1);
270             id_caso_zero = StringUtils.leftPad(id_caso, 10, "0");
271             sql_oracle = "select b.s_title estado "
272                 + "from table_interact i "
273                 + "left join table_case ca on i.x_interact2case= ca.objid "
274                 + "left join sa.table_gbst_elm b on b.objid = ca.casests2gbst_elm "
275                 + "where ca.id_number = '"+id_caso_zero+"' ";
276             rs_oracle = stmt_oracle.executeQuery(sql_oracle);
277             while(rs_oracle.next()){
278                 estado = rs_oracle.getString(1);
279                 stmt_mysql2.execute("update Prueba.Variacion "
280                     + "set estado = " + GetNull(estado) + " "
281                     + "where id_caso = "+id_caso);
282             }
283         }

```

```

279      stmt_mysql2.execute("update Prueba.Variacion "
280      + "set estado = " + GetNull(estado) + " "
281      + "where id_caso = " + id_caso);
282    }
283    i++;
284    if( i % 20000 == 0 )
285      Log.info("Procesando registros : " + i + " registros ... ");
286  }
287  Log.info("En tabla Prueba.Variacion se modificaron " + i + " registros");
288 }catch(Exception e){
289   e.printStackTrace();
290   Log.error("Mensaje de la excepcion : " + e.getMessage());
291   StackTraceElement[] stack = e.getStackTrace();
292   Log.error("Error de Clase : " + e.getClass());
293   for( StackTraceElement ste : stack ){
294     Log.error( ste );
295   }
296 }finally {
297   if (rs_oracle != null)
298     try {
299       rs_oracle.close();
300     } catch (Exception e) {
301     }
302   if (stmt_oracle != null)
303     try {
304       stmt_oracle.close();
305     } catch (Exception e) {
306     }
307   if (conn_mysql != null)
308     try {
309       conn_mysql.close();
310     } catch (Exception e) {
311     }
312 }
313 }
314 }
315 }
316 }
317 }

```

Procedure: Variacion_temp()

```

1 BEGIN
2   declare v_UBICACIONENLARED, n_UBICACIONENLARED, v_mac_cm, v_ts,
3   v_cmts, v_idclienteCRM, v_nodo_correcto, v_obs, v_d1, v_nd2,
4   v_nodosga1, v_nodosga2, v_nodosga, v_plano, v_nc_t, v_nc_t1,
5   v_nc_t2, v_nd3, v_nc_t3, p_s, v_cmtscomp, v_nodo_despues,
6   v_nodo_antes, v_alarma, v_estado, v_ultCola, vCola, v_coment, v_ultimaCola varchar(4000);
7   DECLARE v_item, pos0, cont1, cont2, count2, count3, count, done,
8   num_rows, v_codcli, v_customer_id, v_aux, cont3, cont4, size_nodoc,
9   cont5, cont6, cont7, v_plano1, cont8, v_itemAN, cont9, count1 int DEFAULT 0;
10  declare v_id_interac bigint;
11  declare v_id bigint;
12  declare v_fc, v_fecha_cierre, entry_time_900, entry_time_4100 datetime;
13  DECLARE no_more_rows BOOLEAN;
14  DECLARE duplicate_key CONDITION FOR 1062;
15
16  DECLARE cur1 CURSOR FOR select distinct id_interaccion from Prueba.Variacion_temp
17  ;
18
19  DECLARE CONTINUE HANDLER FOR duplicate_key
20  BEGIN
21    SET count=0;
22  END;
23  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
24  SET no_more_rows = TRUE;
25  OPEN cur1;
26  select FOUND_ROWS() into num_rows;
27
28  insert into AYC_Fija.ControlProcedure(evento, descripcion, fechahora, user )
29  values ('Variacion_temp', 'Variacion_Temp inicio', now(), user());

```

```

29 values ('Variacion_temp','Variacion_Temp inicio',now(),user());
30
31 truncate Prueba.InteracCola;
32
33 the_loop: LOOP
34     FETCH curl INTO v_id_interac ;
35     IF done THEN
36         LEAVE the_loop;
37     END IF;
38     select count(1) into count1 from Prueba.Variacion
39     where id_interaccion = v_id_interac ;
40
41     -- count1 = 0 means there are no records with this ID
42     select count(1) into count2
43     from Prueba.Variacion_temp
44     where id_interaccion = v_id_interac
45     and act_code = 900
46     and INSTR(addnl_info,'predet') > 0 ;
47
48     -- count2 = 0 means there are no records with this ID
49     -- and act_code = 900 and INSTR(addnl_info,'predet') > 0
50     -- and act_code = 900 and INSTR(addnl_info,'predet') > 0
51     -- and act_code = 900 and INSTR(addnl_info,'predet') > 0
52     if ( count2 > 0 ) then
53         select entry_time, id into entry_time_900, v_id
54         from Prueba.Variacion_temp
55         where id_interaccion = v_id_interac
56         and act_code = 900
57         and entry_time = (
58
59         and act_code = 900
60         and entry_time = (
61             select max(entry_time)
62             from Prueba.Variacion_temp
63             where id_interaccion = v_id_interac
64             and act_code = 900
65             and INSTR(addnl_info,'predet') > 0
66             limit 1
67         )
68         and INSTR(addnl_info,'predet') > 0
69         limit 1;
70
71         select addnl_info into v_coment
72         from Prueba.Variacion_temp
73         where id = v_id ;
74
75         set v_ultima_cola = SUBSTR(v_coment, INSTR(v_coment,'cola')+5,900);
76     else
77         set v_ultima_cola = NULL;
78     end if;
79
80     select distinct estado into v_estado
81     from Prueba.Variacion_temp
82     where id_interaccion = v_id_interac
83     limit 1;
84
85     -- count1 = 0 means there are no records with this ID
86     -- count2 = 0 means there are no records with this ID
87     -- count2 = 0 means there are no records with this ID
88     -- count2 = 0 means there are no records with this ID
89     -- count2 = 0 means there are no records with this ID
90     -- count2 = 0 means there are no records with this ID
91     -- count2 = 0 means there are no records with this ID
92     -- count2 = 0 means there are no records with this ID
93     -- count2 = 0 means there are no records with this ID
94     -- count2 = 0 means there are no records with this ID
95     -- count2 = 0 means there are no records with this ID
96     -- count2 = 0 means there are no records with this ID
97     -- count2 = 0 means there are no records with this ID
98     -- count2 = 0 means there are no records with this ID
99     -- count2 = 0 means there are no records with this ID
100    -- count2 = 0 means there are no records with this ID

```

```

89
90     select count(1) into count3
91     from Prueba.Variacion_temp
92     where cola is not null
93     and id_interaccion = v_id_interac;
94     if (count3 > 0) then
95         select distinct cola into vCola
96         from Prueba.Variacion_temp
97         where id_interaccion = v_id_interac limit 1;
98
99
100
101     end if;
102
103
104     if ( count1 = 0 ) THEN
105
106         insert into Prueba.Variacion(ID_CASO,id_interaccion,FECHA_CREACION,DNI,phone,FIRST_NAME,
107             LAST_NAME,PRODUCTO,VARIACION,PROBLEMA, ESTADO,servicioafectado,inconveniente,
108             cola,asesor, x_customer_id )
109         select distinct ID_CASO,id_interaccion,FECHA_CREACION,DNI,phone,FIRST_NAME,
110             LAST_NAME,PRODUCTO,VARIACION,PROBLEMA, ESTADO,servicioafectado,inconveniente,
111             cola,asesor , x_customer_id from Prueba.Variacion_temp where id_interaccion = v_id_interac ;
112         update Prueba.Variacion
113         set
114             estado = v_estado,
115             ultimaCola = v_ultimaCola
116         where id_interaccion = v_id_interac;
117     else
118

```

```

116         where id_interaccion = v_id_interac;
117     else
118
119         update Prueba.Variacion
120         set
121             estado = v_estado ,
122             ultimaCola = v_ultimaCola
123         where id_interaccion = v_id_interac;
124     end if;
125
126     END LOOP the_loop;
127     CLOSE curl;
128
129
130     insert into ATC_Fija.ControlProcedure(evento, descripcion, fechahora, user )
131     values ('Variacion_temp','fin procedure carga Tabla Variacion',now(),user());
132
133
134     insert into ATC_Fija.ControlProcedure(evento, descripcion, fechahora, user )
135     values ('Variacion_temp','Inicio Variacion_P_Table',now(),user());
136
137
138     call ATC_Fija.VariacionP_Table();
139
140
141     insert into ATC_Fija.ControlProcedure(evento, descripcion, fechahora, user )
142     values ('Variacion_temp','Fin Variacion_P_Table',now(),user());
143
144

```



```

197 call ATC_Fija.UpdPrioridadCaso();
198
199 insert into ATC_Fija.ControlProcedure(evento, descripcion, fechahora, user )
200 values ('Variacion_temp','Fin procedure ATC_Fija.UpdPrioridadCaso',now(),user());
201
202 set @days = 2 ;
203 if( DATE_FORMAT(now(), '%H' ) % 5 = 0 and DATE_FORMAT(now(), '%i') < 15 ) then
204     set @days = 210 ;
205 elseif ( DATE_FORMAT(now(), '%H' ) > 6 and ( DATE_FORMAT(now(), '%i') < 6 or
206         ( DATE_FORMAT(now(), '%i') > 30 and DATE_FORMAT(now(), '%i') < 36 ) ) ) then
207     set @days = 60 ;
208 end if;
209
210 update ATC_Fija.CasosATF as c
211 left join Prueba.Variacion as v
212 on c.id_caso = v.id_caso
213 set c.estado = v.estado,
214 c.fecha_cierre = v.fecha_cierre_caso
215 where c.fecha_creacion > DATE_SUB(CURRENT_DATE(), INTERVAL @days day);
216
217 insert into ATC_Fija.ControlProcedure(evento, descripcion, fechahora, user )
218 values ('Variacion_temp',concat ('Fin Variacion_temp days = ' , @days ) ,now(),user());
219
220 call ATC_Fija.AlertaCaso();
221
222 insert into ATC_Fija.ControlProcedure(evento, descripcion, fechahora, user )
223 values ('Variacion_temp','Fin Variacion_temp',now(),user());
224
225 END

```

ANEXO 2:

```
1 package bsos;
2
3 import java.io.BufferedReader;
15
16 public class HFC_ActSus {
17
18     static Logger Log = Logger.getLogger(HFC_ActSus.class.getName());
19
20     public static void main(String[] args) {
21         Calendar cal = Calendar.getInstance();
22         int hora = 0;
23         while(true){
24             try{
25                 cal = Calendar.getInstance();
26                 hora = cal.get(Calendar.HOUR_OF_DAY);
27                 if( hora >= 0 && hora <= 25 ){
28                     GetHFCAct_Sus();
29                     Thread.sleep(9600000);
30                 }
31             }catch(Exception e){
32                 e.printStackTrace();
33             }
34         }
35     }
36
37     public static void GetHFCAct_Sus() {
38         int i = 0 , coid = 0 , customer_id = 0 , tmcode = 0;
39         String sql_insert= "" ;
40         Connection conn_oracle = null , conn_mysql = null , conn_mysql2 = null;
41         Statement stmt_mysql = null , stmt_mysql2 = null , stmt_oracle = null;
42         CallableStatement cStmt = null;
43         ResultSet rs_oracle_hfc = null;
44         ResultSet rs_oracle_ifi = null;
45         String file_query_hfc = "/data/query/HFC_ActSusp.sql";
46         String file_query_ifi = "/data/query/IFI_ActSusp.sql";
47         String sql_oracle_hfc = getContenido(file_query_hfc);
48         String sql_oracle_ifi = getContenido(file_query_ifi);
49         //log.info("### Programa inicio ### ");
50         //System.out.println(sql_oracle);
```

```

111         Log.info("### Programa fin ### ");
112     }catch(Exception e){
113         Log.error(sql_insert);
114         Log.error("Mensaje de la excepcion : "+e.getMessage());
115         e.printStackTrace();
116         StackTraceElement[] stack = e.getStackTrace();
117         Log.error("Error Class: "+e.getClass());
118         for( StackTraceElement ste : stack )
119             Log.error( ste );
120     }finally {
121         if (conn_oracle != null)
122             try {
123                 conn_oracle.close();
124             } catch (Exception e) {
125             }
126         if (stmt_oracle != null)
127             try {
128                 stmt_oracle.close();
129             } catch (Exception e) {
130             }
131         if (conn_mysql != null)
132             try {
133                 conn_mysql.close();
134             } catch (Exception e) {
135             }
136         if (conn_mysql2 != null)
137             try {
138                 conn_mysql2.close();
139             } catch (Exception e) {
140             }
141         if (stmt_mysql != null)
142             try {
143                 conn_mysql2.close();
144             } catch (Exception e) {
145             }
146         if (stmt_mysql2 != null)
147             try {
148                 conn_mysql2.close();
149             }

```

```

152     }
153 }
154
155 private static String FormatoNull(String campo) {
156     String resp="";
157     if(campo==null)
158         resp="NULL";
159     else{
160         if( campo.substring(0, 1).equals(":") )
161             campo = campo.replace(":", "");
162         resp=""+campo+"";
163     }
164     return resp;
165 }
166
167 public static String getContenido(String archivo) {
168     String contenido = "";
169     String cadena;
170     try{
171         FileReader f = new FileReader(archivo);
172         BufferedReader b = new BufferedReader(f);
173         while( ( cadena = b.readLine() ) != null) {
174             contenido = contenido +cadena + "\n";
175         }
176         b.close();
177     }catch(Exception e){
178         e.printStackTrace();
179     }
180     return contenido;
181 }
182
183 }
184 }
185

```

Archivo: /data/query/HFC_ActSusp.sql

```

[USRCSSF@limappnavicv01 query]$ more HFC_ActSusp.sql
select a.co_id, a.customer_id, a.tcode, ch.ch_status , ch.ch_validfrom , cu.passportno dni,
(select d.dn_num from tim.pp_datos_contrato d
where rownum = 1
and d.co_id = b.co_id ) numero
FROM CONTRACT_ALL a , curr_co_status b,contract_history ch, customer_all cu
WHERE ( a.plcode = 1000
or a.tcode in (2580,200,1817,1489,2322,2323,2324,2325,2326,2327,2536
,2537,2539,2566,2567,2580,2581,2582,2583,2586,2587,2588,2589,2598,2604
,2606,2624,2625,2539,2327,2547,2567,2537,2326,2546,2566,2536,2323,2545 ) )
AND a.co_id = b.co_id
AND ch.ch_status in ('a','s')
and a.co_id = ch.co_id
and cu.customer_id = a.customer_id
and ch.ch_seqno = ( select max(h.ch_seqno) from contract_history h
where h.co_id = ch.co_id )

```

```

[USRCSSF@limappnavicv01 query]$ more IFI_ActSusp.sql
SELECT A.CO_ID,
A.CUSTOMER_ID,
A.TCODE,
CH.CH_STATUS,
CH.CH_VALIDFROM,
CU.PASSPORTNO DNI,
(SELECT D.DN_NUM
FROM TIM.PP_DATOS_CONTRATO D
WHERE ROWNUM = 1
AND D.CO_ID = B.CO_ID) NUMERO
FROM CONTRACT_ALL A, CURR_CO_STATUS B, CONTRACT_HISTORY CH, CUSTOMER_ALL CU
WHERE A.TCODE IN ( SELECT TO_NUMBER(VALORI) FROM TIM.TIM_PARAMETROS P WHERE P.CAMPO IN ('PLANES_OLO') )
AND A.CO_ID = B.CO_ID
AND CH.CH_STATUS IN ('a', 's')
AND CH.CH_VALIDFROM < TRUNC(SYSDATE) - 1
AND A.CO_ID = CH.CO_ID
AND CU.CUSTOMER_ID = A.CUSTOMER_ID
AND CH.CH_SEQNO = (SELECT MAX(H.CH_SEQNO) FROM CONTRACT_HISTORY H WHERE H.CO_ID = CH.CO_ID)
[USRCSSF@limappnavicv01 query]$

```

ANEXO 3:

```
1 package test;
2
3 import java.io.BufferedReader;
4
5
6
7
8
9
10
11
12
13
14
15 public class TestTable {
16
17
18     public static void main(String[] args){
19         Connection conn_oracle = null;
20         Connection conn_mysql = null;
21         Statement stmt_oracle = null;
22         String sql_oracle = "", tables="";
23         ResultSet rs_oracle = null;
24         ResultSetMetaData rsmd = null;
25         String opc1 = "", column_reg = "", column_name = "", file_query = "", reg = "";
26         try{
27             if(args.length > 0 ){
28                 opc1 = args[0];
29                 if ( args[0] != null & args.length >= 1 ){
30                     if( opc1.equals("SGA")){
31                         conn_oracle = OracleConn.conectar();
32                     } else if( opc1.equals("BSCS")){
33                         conn_oracle = OracleConnBSCS.conectar();
34                     } else if( opc1.equals("TIMPROD")){
35                         conn_oracle = OracleConnTIMPROD.conectar();
36                     } else if( opc1.equals("DWO")){
37                         conn_oracle = OracleConnDWO.conectar();
38                     } else if( opc1.equals("SANLUIS")){
39                         conn_oracle = ConnMYSQL.conectarOperaciones();
40                     }
41                     stmt_oracle = conn_oracle.createStatement();
42                 }
43                 if ( args[0] != null & args.length == 2 ){
44                     sql_oracle = " select * from " + args[1] + " where rownum < 2 ";
45                     //sql_oracle = " select * from solot where rownum = 1 ";
46                     rs_oracle = stmt_oracle.executeQuery(sql_oracle);
47                     System.out.println("Acceso correcto a tabla: " + args[1] );
48                     System.out.println(" -- ");
49                 }
50                 // Ver todas las tablas que posee el usuario consultado
```

```

49     }
50     // Ver todas las tablas que posee el usuario consultado
51     else if( args[0] != null & args.length == 1 ){
52         sql_oracle = " select owner,object_name, object_type from all_objects "
53             + "where owner not in ('SYS','ORDSYS','OLAPSYS',"
54             + "'MDSYS','EUL4','EUL5','WM SYS','CTXSYS',"
55             + "'SYS','SYSTEM',"
56             + "'MDSYS','QUEST','EUL4','EUL5','ADMINJOB',"
57             + "'COLLECTIONS','OLAPSYS','CTXSYS','EXFSYS')";
58         rs_oracle = stmt_oracle.executeQuery(sql_oracle);
59         while(rs_oracle.next()){
60             tables = rs_oracle.getString(1) + "." + rs_oracle.getString(2)
61                 + " ( " + rs_oracle.getString(3) + " ) ";
62             System.out.println(tables);
63         }
64         System.out.println("Accesos a la base : "+opc1 );
65     }
66     if( args[0] != null & args.length == 3 ){
67         System.out.println("Ejecucion de query custom....." );
68         file_query = args[2];
69         sql_oracle = getContenido(file_query);
70         rs_oracle = stmt_oracle.executeQuery(sql_oracle);
71         rsmd = rs_oracle.getMetaData();
72         int numberOfColumns = rsmd.getColumnCount();
73         System.out.println("Cantidad de columnas = " + numberOfColumns);
74         for (int i = 1; i <= numberOfColumns ; i++) {
75             column_name = rsmd.getColumnName(i);
76             if(i==numberOfColumns)
77                 column_reg = column_reg + column_name ;
78             else
79                 column_reg = column_reg + column_name + "|";
80         }
81         System.out.println(column_reg + "\n");
82         //System.out.println("\n");
83         reg = "";
84         while(rs_oracle.next()){
85             reg = "";
86             for(int i=1; i<=numberOfColumns ; i++) {
87                 if(i==numberOfColumns)
88                     reg = reg + rs_oracle.getString(i) ;
89                 else
90                     reg = reg + rs_oracle.getString(i) + "|";
91             }
92             System.out.println(reg);
93         }
94         //System.out.println("Se ejecutó query sin problemas.");
95     }
96 }
97
98 if ( args.length == 0){
99     System.out.println("Sintaxis: java -jar test_table.jar [opc1] [opc2] [opc3]");
100     System.out.println("opc1: SGA | BSCS | TIMPROD");
101     System.out.println("opc2: objeto de base de datos.");
102     System.out.println("Por ejemplo : java -jar test_table.jar SGA solot ");
103     System.out.println("Si desea ver el acceso a todos los objetos de una "
104         + "base: java -jar test_table.jar [SGA|BSCS|TIMPROD] . "
105         + "Por ejemplo: java -jar test_table.jar BSCS");
106     System.out.println("opc3: si un query se ejecuta sin problemas, para validar ejecución, no data");
107     System.out.println("Por ejemplo : java -jar test_table.jar SGA query \"select count(1) from tiptrabajo\" ");
108 }
109 }catch(Exception e){
110     e.printStackTrace();
111     if ( args.length == 2 ){
112         System.out.println("Acceso denegado a: " + args[1] );
113     }
114 }finally {
115     if (stmt_oracle != null)
116         try {
117             stmt_oracle.close();
118         } catch (Exception e) {
119         }
120     if (conn_oracle != null)
121         try {
122             conn_oracle.close();
123         } catch (Exception e) {
124         }
125     if (rs_oracle != null)
126         try {

```

```

113     }
114     }finally {
115         if (stmt_oracle != null)
116             try {
117                 stmt_oracle.close();
118             } catch (Exception e) {
119             }
120         if (conn_oracle != null)
121             try {
122                 conn_oracle.close();
123             } catch (Exception e) {
124             }
125         if (rs_oracle != null)
126             try {
127                 rs_oracle.close();
128             } catch (Exception e) {
129             }
130     }
131 }
132
133 public static String getContenido(String archivo) {
134     String contenido = "";
135     String cadena;
136     try{
137         FileReader f = new FileReader(archivo);
138         BufferedReader b = new BufferedReader(f);
139         while( ( cadena = b.readLine() ) != null) {
140             contenido = contenido +cadena + "\n";
141         }
142         b.close();
143     }catch(Exception e){
144         e.printStackTrace();
145     }
146     return contenido;
147 }
148
149
150 }
151

```

```

152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

1 BEGIN
2
3     insert into ATC_Fija.ControlProcedure(evento, descripcion, fechahora, user )
4     values ('ContactoR2','inicio ContactoR2',now(), user());
5
6     call Prueba.ContactoR_upd();
7
8     insert into ATC_Fija.ControlProcedure(evento, descripcion, fechahora, user )
9     values ('ContactoR2','fin ContactoR_upd',now(), user());
10
11     update ContactoR_temp
12     set numcomcli = TRIM(numcomcli);
13
14     -- Prueba.ContactoR2
15
16     -- Prueba.ContactoR2
17     -- Prueba.ContactoR2
18
19     insert into ATC_Fija.ControlProcedure(evento, descripcion, fechahora, user )
20     values ('ContactoR2','fin insert ContactoR',now(), user());
21
22     update Prueba.ContactoR
23     set celular = ATC_Fija.getNumContactoR(numcomcli)
24     where INSTR(ATC_Fija.getNumContactoR(numcomcli), '9' ) = 1
25     and LENGTH(ATC_Fija.getNumContactoR(numcomcli)) = 9 ;
26
27     delete c1
28     from Prueba.ContactoR as c1
29     left join ATC_Fija.CLIENTESXPLANO_r as c2

```

```

28     from Prueba.ContactoR as c1
29     left join ATC_Fija.CLIENTESXPLANO_r as c2
30     on c1.codcli = c2.codcli
31     where ISNULL(c2.codcli) = true ;
32
33     insert into ATC_Fija.ControlProcedure(evento, descripcion, fechahora, user )
34     values ('ContactoR2','fin update ContactoR',now(), user());
35
36     truncate table Prueba.ContactoR2 ;
37
38     insert into Prueba.ContactoR2
39     select nro_documento,CELULAR
40     from
41     (
42         select distinct nro_documento
43         ,case
44             when INSTR(celular,'519' ) = 1 then celular
45             when INSTR(celular,'19' ) = 1 then CONCAT('5' , celular)
46             when INSTR(celular,'9' ) = 1 then CONCAT('51' , celular)
47             else ''
48         end as celular
49         from
50         SeguimientoServiciosFijos.MasivoAux
51
52         UNION
53
54         select distinct nro_documento , celular2 from
55         SeguimientoServiciosFijos.MasivoAux

```



```

55      SeguimientoServiciosFijos.MasivoAux
56      where
57      celular2 regexp '^[5][1][9]. '
58
59      UNION
60
61      Select nro_documento, concat('51',celular) celular
62      from Prueba.ContactoR
63      where celular <> ''
64      and celular regexp '^[9]. '
65
66      UNION
67
68      select distinct nro_documento, concat('51', obs_cel )
69      from ATC_Fija.SOI_SGA
70      where fec_gen > DATE_SUB(date(now()),INTERVAL 5 month )
71      and obs_cel regexp '^[9][0-9]. '
72
73      UNION
74
75      select distinct v1.dni, concat('51',v1.celular )
76      from ATC_Fija.VarCelObs as v1
77      inner join Prueba.Variacion_C as v2
78      on v1.id_interaccion = v2.id_interaccion
79      where v1.dni <> '88888888'
80      and v1.celular <> ''
81      and LENGTH(v1.celular) = 9
82      and LENGTH(v1.dni) = 8
83      and v2.fecha_creacion > DATE_SUB(date(now()),INTERVAL 6 month )

```

```

82         and LENGTH(v1.dni) = 8
83         and v2.fecha_creacion > DATE_SUB(date(now()), INTERVAL 6 month )
84         and v1.celular regexp '^[5][1][9].'
85
86     UNION
87
88     select distinct c.documento
89     ,   concat('51',n.numcel)
90     from ATC_Fija.NumWhatsapp as n
91     left join ATC_Fija.CLIENTESXPLANO_r as c
92     on c.codcli = n.CODCLI
93     where length(n.numcel) = 9
94     and c.documento <> '88888888'
95     and n.numcel regexp '^[5][1][9].'
96
97 ) a
98 where celular <> ''
99 and nro_documento <> 'null'
100 and celular <> '519999999999'
101 and celular <> '519000000000'
102 and celular regexp '^[5][1][9].';
103
104 delete from Prueba.ContactoR2
105 where nro_documento IN
106 ( select dni from SeguimientoServiciosFijos.COLABORADOR2020
107   where dni is not null ) ;
108
109 truncate Prueba.ContactoR2_nro_documento;
110
111 truncate Prueba.ContactoR2_nro_documento;
112
113 insert into Prueba.ContactoR2_nro_documento
114 ( select distinct nro_documento from Prueba.ContactoR2
115   group by celular
116   having count(1) > 10 ) ;
117
118 delete from Prueba.ContactoR2
119 where nro_documento in
120 ( select nro_documento from ContactoR2_nro_documento ) ;
121
122 insert into ATC_Fija.ControlProcedure(evento, descripcion, fechahora, user )
123 values ('ContactoR2','fin ContactoR2',now(), user());
124
125 truncate Prueba.ContactoR2_int;
126
127 insert into Prueba.ContactoR2_int(nro_documento, celular)
128 select nro_documento, celular from Prueba.ContactoR2;
129
130 insert into ATC_Fija.ControlProcedure(evento, descripcion, fechahora, user )
131 values ('ContactoR2','fin ContactoR2_int',now(), user());
132 EN

```

ANEXO 4:

```

1 BEGIN
2     DECLARE v_mac_add,v_nro_cli,v_servicio,v_hubiw,v_nodoiw,vdosis_est,v_mensa,v_cliente,v_usuario,
3             v_produ,v_canti,v_model_mta,v_des,v_codubi,v_canthp,v_idhub,v_idcmts,v_nomdst,
4             v_cmtsdes,v_abrevhub,v_sot, v_usuario_cierre varchar(1000);
5     DECLARE des_con,v_obs,v_tipificacion,v_motivoreincio,v_tiposot, v_dni , v_first_name , v_last_name,
6             v_phone , vCola , v_producto , v_variacion , v_problema , v_estado , v_asesor ,
7             v_servicioafectado , v_inconveniente , v_plano , v_hub , v_direccion VARCHAR(2000);
8     DECLARE num_rows,count,done,cont1,cont2,cont3,cont4,cont5,dif,difl,codin,band ,v_codcli ,
9             v_count2,v_ins ,count3 , count4 , v_id_caso , v_customer_id , v_x_customer_id INT DEFAULT 0;
10    declare v_id_inter bigint ;
11    DECLARE fec1,fec2, v_sca_dia date;
12    declare v_hora_agenda_llamada time;
13    declare v_sca, v_fecha_creacion , v_fcc datetime ;
14    DECLARE no_more_rows BOOLEAN;
15    DECLARE duplicate_key CONDITION FOR 1062;
16    DECLARE curl CURSOR FOR select distinct v.id_caso ,
17                                     v.id_interaccion ,v.FECHA_CREACION,v.DNI,v.CUSTOMER_ID,v.FIRST_NAME,v.LAST_NAME,v.PHONE,
18                                     v.COLA,v.PRODUCTO,v.VARIACION,v.PROBLEMA,v.ESTADO,
19                                     v.ASESOR asesor_creador,v.servicioafectado,v.inconveniente , v_estado, v_direccion
20                                     ,v.fecha_cierre_caso, v.usuario_cierre_caso, v.x_customer_id
21                                     from Prueba.Variacion as v
22                                     where
23                                     v.fecha_creacion >= DATE_sub(date(now()),INTERVAL v_param_month month )
24                                     and id_caso is not null
25                                     and ( ( variacion = 'VARIACIÓN - FALLAS TÉCNICAS'
26                                     or (variacion = 'VARIACIÓN - SERVICIOS'
27                                     and problemas= 'SOLICITUDES TÉCNICAS' ) )

```

Parameter:

IN v_param_month int(18)

```

28 and ( ( ( variacion = 'VARIACIÓN - FALLAS TÉCNICAS'
29 or (variacion = 'VARIACIÓN - SERVICIOS'
30 and problema= 'SOLICITUDES TÉCNICAS' ) )
31
32 and ( v.cola = 'ATENCIÓN TECNOLÓG. FIJA' OR
33 v.ultima_cola = 'ATENCIÓN TECNOLÓG. FIJA.'
34
35 ) )
36 OR (
37 v.variacion = 'VARIACIÓN - FALLAS TÉCNICAS'
38 and v.problema = 'CASO - FALLA TÉCNICA'
39 and v.cola = 'ATENCIÓN TECNOLÓGICA'
40 and v.servicioafectado = 'Cuadrilla Especializada'
41 ) ) ;
42
43 DECLARE CONTINUE HANDLER FOR duplicate_key
44 BEGIN
45 SET count=0;
46
47 END;
48 DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
49 SET no_more_rows = TRUE;
50
51 OPEN curl;
52 SELECT FOUND_ROWS() INTO num_rows;
53 the_loop: LOOP
54 FETCH curl INTO v_id_caso, v_id_inter, v_fecha_creacion, v_dni, v_customer_id, v_first_name,
55 v_last_name, v_phone, v_cola, v_producto, v_variacion, v_problema, v_estado,
56 v_asesor, v_servicioafectado, v_inconveniente
57
58 , v_fcc, v_usuario_cierre, v_x_customer_id ;
59
60 IF done THEN
61 LEAVE the_loop;
62 END IF;
63
64 select count(1) into cont2
65 from ATC_Fija.CasosATF
66 where id_caso = v_id_caso ;
67
68 if( cont2 = 0 ) THEN
69 insert into ATC_Fija.CasosATF ( id_caso, id_interaccion, fecha_creacion, dni, customer_id, first_name,
70 last_name, phone, cola, producto, variacion, problema, Estado, asesor_creador, servicioafectado, inconveniente,
71 plano, hub, direccion )
72 values ( v_id_caso, v_id_inter, v_fecha_creacion, v_dni,
73 if(v_producto='DTH POST',v_x_customer_id,v_customer_id )
74 ,v_first_name,
75 v_last_name, v_phone, v_cola, v_producto, v_variacion, v_problema, v_estado,
76 v_asesor, v_servicioafectado, v_inconveniente
77 , ATC_Fija.getDataCap('plano',customer_id)
78 , ATC_Fija.getDataCap('hub',customer_id)
79 , ATC_Fija.getDataCap('direccion',customer_id)
80 ) ;
81
82 else
83 update ATC_Fija.CasosATF
84 set estado = v_estado
85 ,cola = v_cola
86 ,caso_duplicidad = getCasoDuplicidad(v_customer_id, v_id_caso)
87 where id_caso = v_id_caso ;
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

ANEXO 5:

```
select 'a'.id_caso AS id_caso, (case when ('a'.ESTADO = 'PENDIENTE') then timediff(now(), 'a'.FECHA_CREACION) else '' end) AS T_restante_atencion, 'a'.
'id_interaccion' AS id_interaccion, 'a'.FECHA_CREACION AS FECHA_CREACION, 'a'.DNI AS DNI, 'a'.CUSTOMER_ID AS CUSTOMER_ID, 'a'.FIRST_NAME AS
FIRST_NAME, 'a'.LAST_NAME AS LAST_NAME, 'a'.PHONE AS PHONE, 'a'.COLA AS COLA, 'a'.PRODUCTO AS PRODUCTO, 'a'.VARIACION AS VARIACION, 'a'.PROBLEMA
AS PROBLEMA, 'a'.ESTADO AS ESTADO, 'a'.Ofrecimiento de herramienta AS Ofrecimiento de herramienta, 'a'.Servicio afectado videollamada AS
Servicio_afectado_videollamada, 'a'.Aceptacion_cliente AS Aceptacion_cliente, 'a'.Motivo de no aceptacion AS Motivo de no aceptacion, 'a'.NUMERO AS
NUMERO, 'a'.MARCA_MODELO AS MARCA_MODELO, 'a'.SISTEMA_OPERATIVO AS SISTEMA_OPERATIVO, 'a'.OPERADOR AS OPERADOR, 'a'.cuenta_con_smartHome AS
cuenta_con_smartHome, 'a'.se_instala_SmartHome_en_llamada AS se_instala_SmartHome_en_llamada, 'a'.motivo_no_instalacion AS motivo_no_instalacion, 'a'.
asesor_creador AS asesor_creador, 'a'.servicioafectado AS servicioafectado, 'a'.inconveniente AS inconveniente, 'a'.estado_revision AS
estado_revision, 'a'.condicion AS condicion, 'a'.diagnostico AS diagnostico, 'a'.caso_duplicidad AS caso_duplicidad, 'a'.caso_duplicidad_hist AS
caso_duplicidad_hist, 'a'.accion AS accion, 'a'.direccion AS direccion, 'a'.codigo_asesor_asignado AS codigo_asesor_asignado, 'a'.
nombre_asesor_asignado AS nombre_asesor_asignado, 'a'.sot_asociado AS sot_asociado, 'a'.asesor_ult_cola AS asesor_ult_cola, 'a'.alerta AS alerta, 'a'.
Ticket_Remedy AS Ticket_Remedy, 'a'.num1 AS num1, 'a'.num2 AS num2, 'a'.num3 AS num3, 'a'.num4 AS num4, 'a'.Reincidente AS Reincidente, 'a'.
Llamada_agendada AS Llamada_agendada, 'a'.telf_servicio AS telf_servicio, 'a'.prioridad AS prioridad, 'a'.obs AS obs, 'a'.fecha_obs AS fecha_obs,
'a'.movilidad AS movilidad, 'a'.estado_bscs AS estado_bscs, 'a'.id_inter_smarthome AS id_inter_smarthome, 'a'.num_whatsapp AS num_whatsapp, 'a'.MAC
AS MAC, 'a'.modelo_equipo AS Modelo_equipo, 'a'.Firmware AS firmware, 'a'.fecha_diagnostico AS fecha_diagnostico, 'a'.revision_dia AS revision_dia
from CasosATF 'a' where (('a'.ESTADO in ('PENDIENTE', 'CERRADO', 'EN PROCESO', 'SOLUCIONADO')) and (((('a'.servicioafectado <> 'Cuadrilla Especializada') or (
isnull('a'.servicioafectado') = 1)) and ('a'.FECHA_CREACION < '2021-05-23')) or (('a'.FECHA_CREACION > '2021-05-23') and ('a'.FECHA_CREACION < '2021-07-09'
)) or (((('a'.servicioafectado <> 'Cuadrilla Especializada') or (isnull('a'.servicioafectado') = 1)) and ('a'.FECHA_CREACION > '2021-07-09')) and ('a'.
CallCenter in ('', 'DYN'));
```

```
select 'a'.id_caso AS id_caso, 'a'.ESTADO AS ESTADO, 'a'.CUSTOMER_ID AS CUSTOMER_ID, 'a'.FECHA_CREACION AS FECHA_CREACION, 'a'.PROBLEMA AS
PROBLEMA, 'a'.PRODUCTO AS PRODUCTO, 'a'.caso_duplicidad AS caso_duplicidad, 'a'.revision_dia AS revision_dia, 'a'.cuenta_con_smartHome AS
cuenta_con_smartHome, 'a'.se_instala_SmartHome_en_llamada AS se_instala_SmartHome_en_llamada, 'a'.motivo_no_instalacion AS motivo_no_instalacion, 'a'.
Ofrecimiento de herramienta AS Ofrecimiento de herramienta, 'a'.Aceptacion_cliente AS Aceptacion_cliente, 'a'.Servicio_afectado_videollamada AS
Servicio_afectado_videollamada, 'a'.Motivo de no aceptacion AS Motivo de no aceptacion, 'a'.NUMERO AS NUMERO, 'a'.MARCA_MODELO AS MARCA_MODELO, 'a'.
SISTEMA_OPERATIVO AS SISTEMA_OPERATIVO, 'a'.OPERADOR AS OPERADOR, 'a'.sot_asociado AS sot_asociado, 'a'.Llamada_agendada AS Llamada_agendada, 'a'.
obs AS obs, 'a'.diagnostico AS diagnostico, 'a'.num_whatsapp AS num_whatsapp, 'a'.modelo_equipo AS Modelo_equipo, 'a'.MAC AS MAC, 'a'.Firmware AS
firmware, 'a'.id_inter_smarthome AS id_inter_smarthome, 'a'.estado_bscs AS estado_bscs, 'a'.alerta AS alerta, 'a'.Ticket_Remedy AS Ticket_Remedy,
'a'.movilidad AS movilidad, 'a'.num1 AS num1, 'a'.num2 AS num2, 'a'.num3 AS num3, 'a'.num4 AS num4, 'a'.telf_servicio AS telf_servicio, 'a'.
PHONE AS PHONE, 'a'.Reincidente AS Reincidente, 'a'.nombre_asesor_asignado AS nombre_asesor_asignado, 'a'.codigo_asesor_asignado AS
codigo_asesor_asignado, 'a'.servicioafectado AS servicioafectado, 'a'.inconveniente AS inconveniente, 'a'.estado_revision AS estado_revision, 'a'.
condicion AS condicion, 'a'.DNI AS DNI, 'a'.FIRST_NAME AS FIRST_NAME, 'a'.LAST_NAME AS LAST_NAME, 'a'.PLANO AS PLANO, 'a'.hub AS hub, 'a'.
direccion AS direccion, 'a'.Averia AS Averia, 'a'.prioridad AS prioridad, 'a'.canal AS canal, 'a'.asesor_creador AS asesor_creador, 'a'.
caso_duplicidad_hist AS caso_duplicidad_hist, 'a'.fecha_diagnostico AS fecha_diagnostico, 'a'.fecha_obs AS fecha_obs, 'a'.fecha_recupera AS
fecha_recupera, 'a'.operacion AS operacion, 'a'.COLA AS COLA, 'a'.VARIACION AS VARIACION, (case when ('a'.ESTADO = 'PENDIENTE') then timediff(now(),
'a'.FECHA_CREACION) else '' end) AS T_restante_atencion, 'a'.id_interaccion AS id_interaccion, 'a'.asesor_ult_cola AS asesor_ult_cola, 'a'.accion AS
accion, 'a'.var_tec_asesor_30d AS var_tec_asesor_30d, 'a'.var_asesor_mta_reinicio_30d AS var_asesor_mta_reinicio_30d, 'a'.Inconveniente_INC AS
Inconveniente_INC, 'a'.fecha_cierre AS fecha_cierre from CasosATF 'a' where (('a'.CallCenter = 'MDY') and ('a'.FECHA_CREACION > '2022-04-18') and (('a'.
servicioafectado <> 'Cuadrilla Especializada') or (isnull('a'.servicioafectado') = 1)) and ('a'.ESTADO not in ('CERRADO', 'SOLUCIONADO'));
```