

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



Facultad de Ciencias y Tecnologías Computacionales

Monitoreo y Gestión de agentes y colas del centro de llamadas de la plataforma Platel

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor(es): Glenda Nubia Grant Varona

Tutor(es):

- ✓ Ing. Ciencias Informáticas Mairena Arzuaga Limonta
- ✓ Ing. Telecomunicaciones y Electrónica. Ronney Zulueta Esquivel

Cotutor(es): Ing. Informático. Faviannys Gámez Abat.

Consultante(s): Ing. Ciencias Informáticas. Maday Segreo

La Habana, diciembre de 2023

“2023”

Declaración de autoría

Declaro por este medio que yo Glenda Nubia Grant Varona, soy el autor del Trabajo de Diploma **“Monitoreo y Gestión de agentes y colas del Centro de Llamadas en la plataforma Platel”** y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración de autoría en La Habana a los _____ días del mes de _____ del año 2023.

Autor:

Glenda Nubia Grant Varona

Firma autor

Tutores:

Ing. Mairena Arzuaga Limonta
Firma tutor

Ing. Ronney Zulueta Esquivel
Firma tutor



“El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando; lo que más estamos sembrando son oportunidades a la inteligencia”

Fidel

Castro

Dedicatoria

Esta tesis está dedicada a mi mamá, mi papá y a mi familia en general ya que han sido mi motor impulsor, mis guías, siempre han estado conmigo y me han apoyado en los momentos más difíciles a lo largo de mi vida. Los amo.

Agradecimientos

Primeramente, quiero agradecerle a mi Dios y Padre, que siempre me ha guiado, amado y me ha dado fuerzas durante todo el transcurso de mi vida.

Verdaderamente, las palabras no me alcanzan para agradecerle a ese Padre tan justo y Fiel.

A mis padres Eddy y Misleidys por su apoyo incondicional, por estar ahí cada vez que los he necesitado, por darme todo sin esperar nada a cambio, los quiero con la vida.

A mis hermanos David, Daniel y Jonatan porque son parte de mi vida y lo han sido desde el comienzo de mi carrera.

A mis Abuelos Juan Martín y Sarah que han sido siempre como mis segundos padres.

A mi tía Rosa Varona que, aunque no esté cerca de mí en estos momentos, lo está en el corazón, ya que ella ha sido un apoyo incondicional dándome aliento en los momentos en los que he tenido deseos de no seguir.

A mi tía Miriam Varona por sus consejos que siempre me han servido para bien.

A mi tía Glenda Grant que me ha apoyado mucho tanto en lo profesional como en la vida cotidiana, a ser constante y a siempre ser positiva en todos mis pasos.

A mi amigo Fernando, que me ha brindado siempre su apoyo y demostrado que puedo contar con él para lo que necesite.

A mi amigo Xenofonte por ser buen amigo y hacerme reír en momentos determinados.

A mi amiga Yanety y Yinet por ser buenas compañeras y amigas en todo el proceso de mi vida estudiantil.

A mi amiga Mardelis, Dayelin y Laura por darme momentos buenos de confianza y alegría.

A Anette quien me prestó su laptop para poder trabajar en la tesis y lo hizo en el momento que más lo necesitaba.

A mi tutor Ronney que confió en mí para esta tarea, me apoyó siempre que lo necesité.

A mi tutora Mairena por su apoyo y su ayuda en el proceso de la tesis.

A Faviannys y a Maday que fueron como unos tutores más y me han guiado en el transcurso de este proyecto.

A los profesores de la Facultad y los trabajadores del centro.

A todas las personas que de una forma u otra ayudaron a la realización de este trabajo.

Resumen

La Plataforma Integral de Telecomunicaciones (PLATEL) es la solución nacional capaz de unificar el soporte de hardware y software mediante un conjunto de tecnologías basadas en software libre para brindar una solución integral de comunicaciones seguras para las entidades cubanas con una alta soberanía e independencia tecnológica de acuerdo a los intereses del cliente. Con el objetivo de lograr elevar la calidad de los servicios brindados y agilizar la comunicación masiva con los clientes, se han implantado Centros de Atención de Llamadas.

Para elevar el desempeño llevado a cabo en estos Centros de Atención de Llamadas, se requiere el desarrollo de un sistema que brinde la posibilidad de supervisar y monitorear concurrentemente los diferentes eventos que tienen lugar en los mismos. Se necesita una aplicación que permita visualizar el estado de las llamadas, agentes y colas, así como monitorear las actividades que se desarrollen en el Centro de Atención de Llamadas. El presente trabajo permite la creación de un sistema de monitoreo y gestión que permite monitorear las colas, y agentes del Centro de Atención de Llamadas de la plataforma Platel.

PALABRAS CLAVES:

Agente, Freeswitch, Centro de Atención de Llamadas, cola de llamada, agente y monitoreo y gestión

Abstract

The Comprehensive Telecommunications Platform (PLATEL) is the national solution capable of unifying hardware and software support through a set of technologies based on free software to provide a comprehensive secure communications solution for Cuban entities with high sovereignty and technological independence. according to the client's interests. With the objective of raising the quality of the services provided and streamlining mass communication with clients, Call Centers have been implemented.

To increase the performance carried out in these Call Centers, the development of a system is required that provides the possibility of concurrently supervising and monitoring the different events that take place in them. An application is needed that allows you to view the status of calls, agents and queues, as well as monitor the activities that take place in the Call Center. The present work allows the creation of a monitoring and management system that allows monitoring the queues, and agents of the Call Center of the Platel platform.

KEYWORDS:

Agent, Freeswitch, Call Center, call queue, agent and monitoring and management

ÍNDICE GENER

Introducción.....	xi
Capítulo 1. Fundamentación teórica.....	1
1.1 Conceptos asociados al tema.....	1
1.2 Principales cambios tecnológicos en la industria de las telecomunicaciones.....	3
1.2.1 Red de telefonía tradicional o fija.....	3
1.2.2 Telefonía móvil.....	3
1.2.3 Internet.....	3
1.2.4 Voz sobre el protocolo internet (VoIP).....	3
1.2.5 Comparación Voz sobre IP y telefonía fija.....	4
1.3 Comunicaciones Unificadas.....	5
1.3.1 Platel.....	5
1.3.1.1 Centro de Llamadas.....	6
1.4 Estudio de Soluciones Existentes.....	9
1.5 Metodología utilizada en la solución propuesta.....	10
1.6 Tecnologías y herramientas asociadas al desarrollo del sistema.....	12
1.6.1 Herramientas de modelado.....	12
1.6.2 Gestor de bases de datos y Base de datos.....	13
1.6.3 Entorno de Desarrollo Integrado (IDE).....	13
1.6.4 Tecnologías.....	14
1.6.5 Lenguajes de programación.....	15
1.7 Conclusiones parciales.....	16
Capítulo 2: Propuesta de Solución.....	17
2.1 Análisis de la propuesta de solución para gestionar y monitorear los Agentes y Colas del centro de llamadas de la plataforma Platel.....	17
2.1.1 Descripción de la propuesta de solución del sistema.....	18
2.3 Definición de los requisitos de la Aplicación.....	18
2.3.1 Especificación de los requisitos funcionales.....	18
2.3.2 Especificación de requisitos no funcionales.....	22

2.3.3 Descripción de requisitos mediante Historias de Usuarios.....	23
2.4. Arquitectura de software.....	28
2.4.1 Patrones de diseño.....	31
2.5 Modelo de Datos.....	33
2.6 Conclusiones parciales.....	34
CAPÍTULO 3: PRUEBAS O VALIDACIÓN.....	35
3.1 Diagrama de Componentes.....	35
3.2 Diagrama de despliegue.....	36
3.3 Pruebas.....	37
3.4 Pruebas de Integración.....	38
3.5 Pruebas Funcionales.....	41
3.6 Pruebas de Aceptación.....	44
3.7 Conclusiones parciales.....	45
Conclusiones generales.....	46
Recomendaciones.....	47
Referencias Bibliográficas.....	48
ANEXOS.....	52

Índice de TablasY

Tabla 1.1 Comparación entre telefonía y voz sobre.....	4
Tabla 1.2 Historia de Usuario (Adicionar Cola).....	11
Tabla 2.1 Requisitos Funcionales.....	22
Tabla 2.2 Historia de Usuario (Autenticar Usuario).....	25
Tabla 2.3 Historia de Usuario (Adicionar Agente).....	26
Tabla 2.4 Historia de Usuario (Adicionar Cola).....	27
Tabla 3.1 Pruebas Realizadas.....	38
Tabla 3.2 Casos de Pruebas.....	39
Tabla 3.3 Casos de pruebas Funcionales.....	43
Tabla 3.4 Casos de pruebas Funcionales.....	43

Índice de Figuras

Figura 1.1. Comunicaciones Unificadas.....	5
Figura 1.2. Funciones que se combinan en los centros de llamadas.....	7
Figura 2.1. Subsistemas de Monitoreo, Supervisión y Gestión.....	17
Figura 2.2. Diagrama de clases del diseño de la Historia de Usuario Adicionar Agente.....	28
Figura 2.3. Capa de Modelo.....	29
Figura 2.4. Capa de Views.....	30
Figura 2.5. Capa de templates.....	30
Figura 2.7. Ejemplo de patrón controlador.....	31
Figura 2.9. Ejemplo de patrón bajo acoplamiento.....	32
Figura 2.10. Ejemplo de Patrón Gof (Decorador).....	33
Figura 2.11. Modelo de Datos.....	33
Figura 3.1. Diagrama de componentes.....	35
Figura 3.2. Diagrama de Despliegue.....	36
Figura 3.3. Vista de cómo se añade el agente en la web.....	39
Figura 3.4. Prueba de Integración para añadir agente.....	40
Figura 3.5. Vista de cómo se añade la cola en la web.....	40
Figura 3.6. Prueba de Integración para añadir cola.....	40
Figura 3.7. Representación gráfica de las pruebas de Integración.....	41
Figura 3.8. No Conformidades de las pruebas Funcionales.....	44

Introducción

El uso de Internet y las Tecnologías de la Información y las Comunicaciones (TICs) ha significado a [escala](#) mundial un salto vertiginoso en el desarrollo científico técnico. Desde su llegada a los escenarios nacionales se han convertido en un elemento indispensable para establecer las líneas de desarrollo de la sociedad cubana, buscando dar solución a los problemas del [hombre](#) [CITATION Yos16 \l 2058].

Las TICs están hondamente arraigadas en el tejido social y forman parte de la manera en que se proyectan los negocios, transformando así el modo de trabajar del ser humano donde es libre de sus cargas más pesadas, optimizando los [recursos](#) y haciendo el mundo digital más productivo [CITATION Yos16 \l 2058]. Una sociedad que aplique la informatización en todas sus esferas y procesos será una sociedad más competitiva.

Con este nuevo auge en la tecnología, las empresas se abren a una nueva forma de relacionarse con sus clientes, en donde la incorporación de servicios de Call Center o centro de llamadas han pasado a ser una de las piezas claves en el conjunto de las gestiones de la calidad de los servicios y la satisfacción de los clientes. También son una efectiva solución de negocios capaz de adaptarse a las cambiantes expectativas de sus clientes, y de esta forma asegurar su lealtad [CITATION CAS091 \l 2058].

Los Call Center se proyectan como una alternativa para que las empresas puedan resolver efectivamente su proceso de toma de pedidos, al integrar a su gestión comercial, tecnología y recursos humanos capaces de llevar a cabo esta función bajo la administración y supervisión constante de un experto.[CITATION CAS091 \l 2058]

En Cuba han surgido entidades que cuentan con una alta responsabilidad en el desarrollo socioeconómico y en la informatización de la sociedad cubana. Dichas entidades son la Empresa de Telecomunicaciones de Cuba (ETECSA), la Universidad de las Ciencias Informáticas (UCI), el Parque Tecnológico de La Habana y otros más.

Con sede en el campus universitario de la UCI se encuentra la Empresa de Tecnologías de la Información para la Defensa, conocida como XETID, fundada en el año [2013](#), dedicada al sector del software, la automática y las comunicaciones. Asume como misión la proyección, diseño, desarrollo y comercialización de productos y servicios, a partir del uso de las tecnologías de la informática y las comunicaciones [CITATION Xet23 \l 2058].

Esta empresa presta servicios técnicos gerenciales y de consultoría informática, despliegues, transferencias de tecnologías, desarrollo de estrategias, canales de información y comunicación; así como soluciones tecnológicas integrales que garantizan y contribuyen al desarrollo del proceso de informatización del país. Con varias sedes en Cuba ofrece soluciones informáticas que abarcan los principales sectores y necesidades

empresariales. Además, brinda un servicio de soporte técnico con cobertura en todo el país, garantizando a sus clientes la estabilidad y mantenimiento de las soluciones [CITATION Xet23 \l 2058].

Con el propósito de brindar servicios de comunicaciones de Voz sobre Protocolo de Internet (VoIP) a las empresas que las solicitan, la XETID implementó una plataforma telefónica PLATEL que tiene como objetivo principal la creación de una plataforma capaz de integrar los servicios de telefonía, correo y mensajería instantánea y así resolver los problemas más destacados en las diferentes entidades del país.

La Plataforma de Telecomunicaciones “Platel” es la integración de tecnologías basadas en software libre y desarrollos propios de XETID para conectar en tiempo real dos o más personas en lugares distintos en un entorno colaborativo que les permita la comunicación segura e intercambiar contenidos multimedia. Ha contribuido al desarrollo de empresas, gobiernos y entidades presupuestadas con el despliegue de aplicativos como Platel-PBX, Platel-Videoconferencia y Platel-Call Center. Platel-Videoconferencia ha sido desplegado en centros de datos de clientes en muchos casos con arquitecturas de despliegue y configuraciones diferentes para adaptarse a las características de las redes de los usuarios.

El impacto de Platel ha sido muy favorable, ya que ha facilitado el teletrabajo y trabajo a distancia de muchas organizaciones, además ha permitido ahorrar tiempo, dinero y combustible. El uso de los servicios de Platel fue la alternativa más viable para mantener niveles de producción en tiempos de pandemia por lo que ha aumentado la demanda.

Platel-Call Center es la solución que permite a los agentes manejar tipos adicionales de contactos de clientes, socios y/o asociados. Esos contactos pueden venir por correo electrónico, llamadas conmutadas públicas, llamadas basadas en Web, por ejemplo: voz sobre protocolo de internet o VoIP (voz sobre ip o telefonía IP como también se le conoce), fax, chat web o cualquier otro canal de comunicación. La solución puede manejar uno, todos o cualquier combinación de esos canales, la cual enruta, pone en cola y gestiona de otra manera cada uno de los canales de contacto.

Esta plataforma posee un núcleo que usa como motor de telefonía Freeswitch¹ que en la actualidad se utiliza como centralita privada, PBX por sus siglas en inglés, la cual se encarga de establecer conexiones entre terminales de una misma empresa, o de hacer que se cursen llamadas al exterior.

¹Es un Softswitch de código abierto a través del cual se pueden crear productos de voz, vídeo y mensajería instantánea. El Softswitch es el principal dispositivo en la capa de control dentro de una arquitectura NGN (Next Generation Network), encargado de proporcionar el control de llamada, procesamiento de llamadas y otros servicios, sobre una red de conmutación de paquetes.

El poder de este motor de telefonía de nueva generación, denominado Freeswitch, está en su capacidad de ser extendido fácilmente. Permite unificar voz, video, datos y mensajería instantánea bajo el mismo motor de enrutamiento, maximizando la eficiencia de las comunicaciones y minimizando los costos de infraestructura en las empresas.

Actualmente en la empresa Xetid, para obtener reportes históricos que ayuden a mejorar los niveles de servicio de Platel y monitorear las colas y agentes es necesario realizarlo a base de comandos. Igualmente, la gestión de los mismos se realiza a través de un archivo XML, donde su manipulación puede llevar a cometer errores. Por lo que resulta tedioso realizar todas estas tareas relacionadas con este servicio. No se logra supervisar de forma concurrente todos los eventos asociados al monitoreo de agentes y colas, por lo que existe un retardo en la visualización de los eventos ocurridos y de esta forma se muestran posibles resultados desactualizados. Por lo tanto, se dificulta el monitoreo de las colas y los agentes asociados a los centros de atención de llamadas. Tampoco es posible visualizar en corto tiempo el comportamiento de las llamadas ni el desempeño de los agentes

El planteamiento anterior constituye el basamento del trabajo que sustenta el presente documento de tesis, realizado para dar respuesta al siguiente **problema a resolver**: ¿Cómo facilitar el monitoreo y gestión de las colas y los agentes en el centro de llamadas de la plataforma PLATEL?

Este problema se enmarca en el **objeto de estudio**: El proceso de monitoreo y gestión de agentes y colas en los centros de atención de llamadas enmarcado en el **campo de acción**: Proceso de monitoreo y gestión de colas y agentes en centros de llamadas de la plataforma PLATEL.

Se traza como **objetivo general**: Desarrollar una aplicación web que permita monitorear y gestionar las colas y agentes en el centro de llamadas de la plataforma PLATEL.

Del objetivo general planteado se desglosan en los siguientes **objetivos específicos**:

- ✓ Elaborar el marco teórico de la investigación para prevenir errores que se han cometido en otros estudios y guiar el desarrollo de la investigación.
- ✓ Diseñar una propuesta de solución que permita facilitar el monitoreo y gestión de las colas y los agentes en el centro de llamadas de la plataforma PLATEL
- ✓ Implementar una aplicación web para el monitoreo y gestión de agentes y colas del centro de llamadas de la plataforma Platel.
- ✓ Validar la aplicación web para comprobar su correcto funcionamiento.

Para guiar el desarrollo del trabajo se plantea el siguiente **resultado**: Aplicación web que permita monitorear y gestionar agentes y colas en tiempo real en la plataforma PLATEL.

Durante el proceso de investigación y elaboración de este trabajo se tomó en cuenta la utilización de varios métodos científicos de investigación:

✓ **Método Teórico**

- o **Analítico-Sintético**: Para analizar las teorías y los documentos referentes a la central telefónica Freeswitch y los Centros de Atención de Llamadas, permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.
- o **Análisis histórico-lógico**: Este método permitirá analizar la trayectoria y evolución del software de Monitoreo y Gestión en las plataformas telefónicas existentes en el mundo y tendencias más actuales lo que servirá para profundizar el conocimiento sobre estos, de tal forma que se pueda conocer de manera internacional el desarrollo de los Centros de Atención de Llamadas.

✓ **Métodos Empíricos:**

- o **Observación**: Para la percepción selectiva de las restricciones y propiedades del sistema y para la determinación de la problemática que da origen a la investigación. La observación que se realiza es estructurada ya que previamente se establecen los aspectos que se desean observar sistematizando los detalles más significativos para la investigación.
- o **Entrevista**: Permite obtener información acerca de la investigación y la realización de un buen levantamiento de requisitos del sistema. Se hizo uso de él para determinar qué funcionalidades se necesitaban e incorporarlas a la aplicación.

El presente trabajo de diploma tiene como estructura capilar la siguiente:

Capítulo 1: Fundamentación teórica

Se especifican los conceptos que son tratados a lo largo del documento y que tienen vital importancia para la comprensión del mismo. Se realiza un estudio de los Distribuidores Automáticos de Llamadas utilizados en Cuba y en el mundo; describiendo las diferentes herramientas, metodologías y lenguajes de programación a utilizar durante la construcción del sistema.

Capítulo 2: Propuesta de solución

Se realiza una descripción detallada de los requisitos funcionales y no funcionales de la aplicación, propiciando una modelación adecuada para la propuesta realizada. Abarca el diseño de cada uno de los artefactos creados en correspondencia con la propuesta de la aplicación, utilizando los diseños y las plantillas especificadas en el Proceso de Desarrollo del centro.

Capítulo 3: Pruebas o Validación

Implementación y validación de la aplicación de Monitoreo y Gestión de Agentes y Colas de un Centro de Llamadas de la plataforma Platel. En este capítulo se construye la propuesta de solución antes descrita. Se presentan las principales tareas de implementación a realizar y los resultados de las diferentes pruebas de software.

Capítulo 1. Fundamentación teórica

Este capítulo tiene como meta principal explicar de forma detallada qué es un Centro de Llamadas y los diversos conceptos básicos relacionados, permitiendo una mejor comprensión del tema. Además, se analizarán las diferentes herramientas, lenguajes y metodologías a usar en el desarrollo de la aplicación web.

1.1 Conceptos asociados al tema

Este epígrafe tiene como objetivo dejar claros los conceptos fundamentales para un mejor entendimiento del dominio del problema.

PBX (Central telefónica privada): Una PBX (Private Branch Exchange por sus siglas en inglés) es una central telefónica privada, que maneja o administra llamadas entrantes y salientes en una empresa que utilizan redes VoIP[CITATION Wal21 \l 2058].

Este constituye la base del sistema de comunicaciones y proporciona la infraestructura necesaria para manejar las comunicaciones telefónicas dentro de una empresa.

Centro de Llamadas: Se trata de la [oficina](#) donde un grupo de personas específicamente entrenadas se encarga de brindar algún tipo de atención o servicio telefónico[CITATION Jul23 \l 23562].

El centro de llamadas utiliza la infraestructura de la PBX para gestionar las llamadas de manera eficiente.

Freeswitch: Es un servidor de telecomunicaciones de código abierto **que proporciona un** entorno completo de PBX. Es conocido por su capacidad de enrutar y gestionar llamadas, y es altamente flexible y personalizada para adaptarse a una variedad de casos de uso y requisitos específicos[CITATION Ant18 \l 3082].

Es importante destacar que Freeswitch es una plataforma esencial para la gestión de llamadas y la interacción con los clientes en un centro de llamadas. Proporciona herramientas necesarias para enrutar, gestionar y personalizar la experiencia del cliente durante las interacciones telefónicas.

Agente: Persona que responde llamadas en un centro de llamada. [CITATION Wal21 \l 2058]

En cuanto a los agentes, la autora considera que desempeñan un papel crucial en la atención al cliente, ya que son la cara de la empresa.

Cola de llamadas: Tipo de número de directorio creado para mantener llamadas o mensajes que están esperando ser atendidas. [CITATION Wal21 \l 2058]

Cabe destacar que la gestión de cola de llamadas es crucial para evitar largos tiempos de espera. Por lo tanto, este ofrece funcionalidades valiosas, como la transferencia de llamadas y la administración de extensiones, que mejoran la comunicación tanto interna como externa de la organización.

Extensión telefónica: Se refiere a la línea conectada a una central telefónica, a la cual se le asigna un número de teléfono para que desde otra línea telefónica al marcar este número se pueda establecer una comunicación entre ambas líneas[CITATION Ari15 \l 2058].

Es decir, la extensión telefónica es el numero al que se dirigen las llamadas que están en espera para ser atendidas por un agente.

ANI (Identificación automática de número): Esta función permite identificar el número del que llaman, también conocido como CallerID [CITATION Wal21 \l 2058].

Por tanto, el ANI puede ayudar a identificar la extensión telefónica desde la cual se realizó la llamada. Es un sistema que identifica automáticamente el número de teléfono del llamante. Este servicio proporciona a la parte receptora de la llamada (como un IVR o un agente humano) información sobre el origen de la llamada.

IVR o respuesta interactiva de voz: Es un sistema de información telefónica automatizada que le habla a la persona que llama con una combinación de menús de voz y datos extraídos de las bases de datos en tiempo real. La persona que llama responde pulsando el teclado del teléfono o pronunciando palabras o frases cortas[CITATION Wal21 \l 2058].

Cuando un llamante realiza una llamada, la interacción puede comenzar con el IVR, que le guiará a través de un menú de opciones a recopilar información inicial y direccionará la llamada o proporcionará información básica antes de que la llamada sea enviada a un agente.

ACD (Distribuidor automático de llamadas): Sistema telefónico que responde llamadas entrantes, obtiene información e instrucciones de una base de datos, determina la mejor forma de procesar la llamada y redirecciona al agente correcto en cuanto está disponible. [CITATION Wal21 \l 2058]

Por lo tanto, una vez que se completa la interacción con el IVR y la llamada es transferida a un agente, el sistema ACD entra en juego. El ACD utiliza métricas tales como disponibilidad y habilidades de los agentes, la prioridad de llamada, la carga de trabajo actual, entre otras estrategias, para enrutar de manera eficiente la llamada al agente mas adecuado para atenderla.

Tiempo de llamada: Indicador de un centro de llamada que determinar el tiempo promedio de atención de una llamada.[CITATION Wal21 \l 2058]

Durante el ciclo completo de la llamada telefónica, el tiempo de llamada es una métrica que se monitorea para evaluar la duración promedio de las llamadas, identificar patrones de uso, determinar los tiempos de espera y evaluar la eficiencia y satisfacción del cliente.

1.2 Principales cambios tecnológicos en la industria de las telecomunicaciones

Las tecnologías de la Información y las Comunicaciones han permitido la creación de nuevas formas de comunicación, como la mensajería instantánea, el correo electrónico y la video-conferencia, las cuales se han vuelto cada vez más populares. En cuanto a los cambios en la industria de las telecomunicaciones, la evolución de la tecnología ha tenido gran impacto en la forma en que las personas se comunican y en como las empresas ofrecen su servicio.

1.2.1 Red de telefonía tradicional o fija

Las primeras centrales telefónicas digitales se introdujeron en la década de 1960 y se desarrolló el concepto de la conmutación de paquetes para la transmisión de datos. Este concepto consiste en que la información es dividida en paquetes, los cuales, una vez enviados desde el equipo emisor, podían tomar diferentes rutas hasta llegar a su destino, el equipo receptor[CITATION Rui23 \l 2058].

1.2.2 Telefonía móvil

Las comunicaciones móviles siguen el principio general de la telefonía: Conectar dos usuarios remotos a través del equipo de red de un operador responsable de la gestión del servicio. Sin embargo, a diferencia de los teléfonos fijos, en la red móvil no existen pares de cobre ni fibra óptica, y las transmisiones de radio constituyen el enlace final[CITATION Jan21 \l 2058].

1.2.3 Internet

Durante los últimos años, Internet ha crecido hasta incluir el potencial informático de las universidades y centros de investigación. Esto, unido a la posterior incorporación de empresas privadas, organismos públicos y asociaciones de todo el mundo, supuso un fuerte impulso para Internet, que dejó de ser un proyecto con protección estatal para convertirse en la mayor red de computadoras del mundo, conformada por un billón de usuarios en enero del 2006[CITATION Rui23 \l 2058].

1.2.4 Voz sobre el protocolo internet (VoIP)

La VoIP es el resultado de la convergencia entre la telefonía e Internet. Como concepto apareció a mediados de la década de 1990 bajo la forma de un programa informático que

permitía transmitir una llamada telefónica por Internet en vez de por medio de la red de telefonía pública conmutada (telefonía tradicional)[CITATION Rui23 \l 2058].

Se puede considerar a la VoIP como una tecnología que está actualmente reformando los modelos de negocios de las empresas operadoras de telecomunicaciones no solamente en términos de los costos de las comunicaciones, sino también en términos de nuevas aplicaciones que facilitan la provisión de nuevos servicios[CITATION Rui23 \l 2058].

1.2.5 Comparación Voz sobre IP y telefonía fija.

Son varias las diferencias entre estos dos sistemas de telefonía. Una de las principales es que la telefonía fija utiliza conmutación de circuitos entre dos puntos, mientras que la voz sobre IP utiliza conmutación de paquetes[CITATION EWE182 \l 2058]. A continuación, se muestra una comparación entre los dos sistemas en la Tabla 1.1.

Telefonía fija	Voz sobre IP
<i>Líneas dedicadas.</i>	Todos los canales llevados a través de una conexión a Internet.
<i>Cada línea es de 64 kbps (en cada dirección).</i>	La compresión puede dar como resultado 10 kbps (en cada dirección).
<i>Características tales como llamada en espera, identificador de llamadas, etc. generalmente están disponibles a un costo adicional.</i>	Las funciones como llamada en espera, identificador de llamadas, etc. normalmente se incluyen de forma gratuita con el servicio.
<i>Se puede actualizar o expandir con nuevos equipos y aprovisionamiento de línea.</i>	Las actualizaciones generalmente solo requieren ancho de banda y actualizaciones de software.
<i>La larga distancia suele ser por minuto o suscripción por minuto.</i>	La larga distancia a menudo se incluye en el precio mensual regular.
<i>Los teléfonos fijos (aquellos sin un adaptador) usualmente permanecen activos durante un apagón.</i>	Pierda potencia, pierda el servicio telefónico sin respaldo de energía en su lugar.
<i>Al realizar una llamada al 911, se puede rastrear hasta su ubicación</i>	Las llamadas de emergencia al 911 no siempre se pueden rastrear a una ubicación geográfica específica.

Tabla 1.1 Comparación entre telefonía y voz sobre

Fuente: Elaboración Propia

1.3 Comunicaciones Unificadas

La evolución de las telecomunicaciones ha llevado una mayor demanda de soluciones de comunicación integradas que permitan a las empresas gestionar y optimizar sus procesos de comunicación. Debido a esto, las comunicaciones unificadas se han convertido en una solución atractiva para las empresas que buscan mejorar la eficiencia y reducir los costos en la gestión de comunicación [CITATION Mor20 \l 2058].

Las soluciones de comunicación unificada surgen como alternativa ideal para aquellas empresas que desean agilizar su proceso de gestión de datos. Son una solución tecnológica que integra diferentes herramientas de comunicación como: los servicios de telefonía, correo electrónico, correo de voz y fax, mensajería instantánea, conferencias web y estado de disponibilidad del usuario en un solo sistema PBX en una única plataforma. Permite una mayor eficiencia en la gestión de la comunicación y una mejor experiencia del usuario [CITATION Mor20 \l 2058].



Figura 1.1. Comunicaciones Unificadas

Fuente: Elaboración propia

En resumen, las Comunicaciones Unificadas son una solución tecnológica que se ha desarrollado en respuesta a los cambios en la industria de las telecomunicaciones, y que permite una mayor eficiencia y flexibilidad en la gestión de la comunicación empresarial.

1.3.1 Platel

La Plataforma Integral de Telecomunicaciones (PLATEL) es la solución nacional capaz de unificar el soporte de hardware y software mediante un conjunto de tecnologías basadas en software libre para brindar una solución integral de comunicaciones seguras para las entidades cubanas con una alta soberanía e independencia tecnológica de acuerdo a los intereses del cliente. Consisten en la convergencia de diferentes módulos que incluyen la telefonía, conferencias, correo electrónico, correo de voz, mensajería instantánea, vídeo y

colaboración a través de una variedad de interfaces, ya sea PC o los clientes basados en la Web, teléfonos y dispositivos móviles, o del habla.

El despliegue de Platel en diferentes empresas es crucial para tener un sistema de comunicaciones efectivo que permita a los trabajadores comunicarse entre sí y atender a los clientes de manera eficiente en el centro de llamadas.

1.3.1.1 Centro de Llamadas

Existen algunas definiciones sobre los centros de llamadas, por ejemplo, Buscatto dice lo siguiente: “basados en la fusión de numerosas revoluciones tecnológicas, los centros de llamadas se definen por estar basados en un mismo principio: la integración de la tecnología y del teléfono para responder en tiempo real a las demandas de los clientes, potenciales o existentes. Las llamadas son recibidas por un sistema informático centralizado que asegura de manera continua y en tiempo real su distribución entre los teleoperadores y el control de la actividad (tiempo de espera, duración de la comunicación, etc.). Los teleoperadores, en su tarea, lo que hacen es utilizar, constituir o transformar una base de datos de los clientes de la empresa”[CITATION Bus02 \l 1033].

Buscatto agrega que los centros de llamadas son extremadamente diversos más allá de su definición formal ya que se encuentran en múltiples sectores como los servicios financieros, las telecomunicaciones, la salud o el sector audiovisual.

McPhail postula que un centro de llamadas es “una oficina especializada donde agentes proveen información, distribuyen servicios y/o llevan adelante ventas usando alguna combinación de tecnología telefónica e informática, típicamente con el objetivo de mejorar la atención al cliente a la vez que con ello se reducen costos organizacionales. [CITATION McP02 \l 1033]

Existen en principio dos funciones en un centro de llamadas: inbound y outbound. En el primer caso las llamadas son entrantes de parte de clientes o potenciales clientes, con el objetivo de que se les provea información, soporte técnico o algún servicio. En el caso outbound, las llamadas son salientes y los agentes llevan adelante tareas como el relevo de datos, recolección de fondos y, sobre todo, la televenta”[CITATION McP02 \l 1033].

En general, los centros de llamadas pueden combinar funciones o servicios outbound e inbound, pero es útil hacer esta distinción, sostiene la autora, porque cada uno implica un tipo de trabajo diferente. A continuación, se muestra en la figura 1.2 un ejemplo de ello.

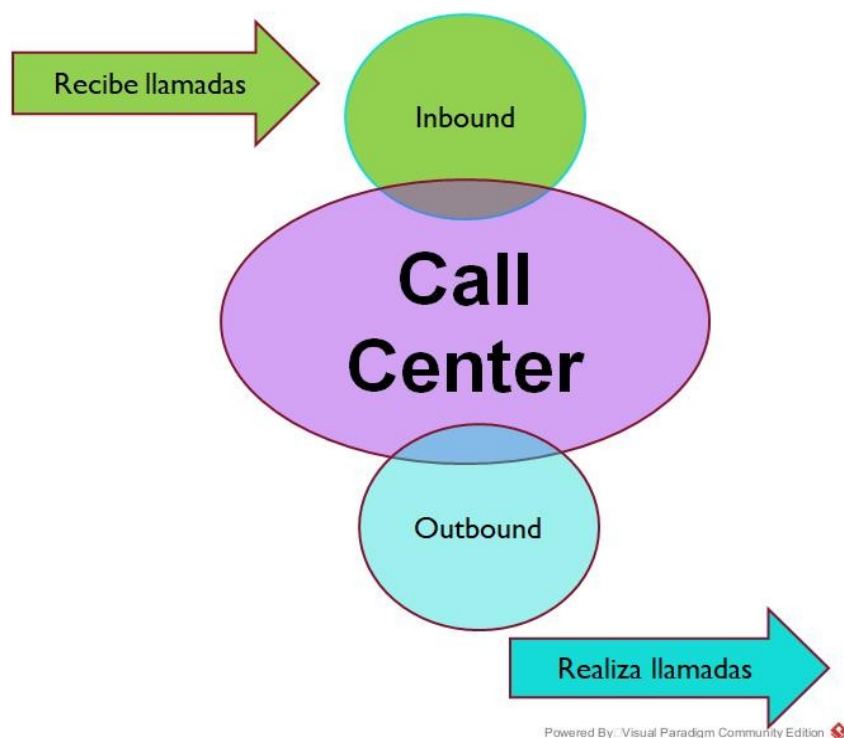


Figura 1.2. Funciones que se combinan en los centros de llamadas

Fuente: Elaboración propia

En la imagen se evidencian los dos tipos de servicios (las llamadas inbound y outbound) esenciales realizadas por la mayoría de equipos de soporte y servicio al cliente. Mientras que las llamadas inbound son iniciadas por el cliente, las outbound se producen cuando el representante es quien llama al cliente primero y funcionan como un soporte extra.

Los centros de llamadas presentan una estructura operativa, expuesta en niveles tales como: gestor, supervisor, asesor y servicios.

El gestor de un centro de llamadas: es la persona que se responsabiliza sobre la planificación estratégica en lo relacionado a la distribución de los supervisores y asesores del servicio y a los parámetros e indicadores de cumplimiento de los servicios que son cubiertos por la central de atención telefónica [CITATION MIC18 \l 2058].

El supervisor: es quien se responsabiliza por el alcance de las metas planteadas para el grupo de asesores que tiene bajo su supervisión, dentro de un determinado periodo de tiempo. Además, son los encargados del perfeccionamiento de la actividad, la calidad de la misma, quienes controlan los costos involucrados y los resultados obtenidos dentro de los centros de llamadas [CITATION MIC18 \l 2058].

El asesor: Empleado que tiene como actividad especializada la atención y gestión con el cliente [CITATION MIC18 \l 2058].

El servicio: Los centros de llamadas pueden combinar funciones o servicios están clasificados bajo dos denominaciones, Inbound (entrada) y Outbound (salida) [CITATION MIC18 \l 2058].

Funciones de los centros de llamadas

Un centro de llamadas tiene como una de sus funciones principales el dar soluciones de información a los clientes, al emplear el contacto telefónico como canal comunicativo. En un centro de gestión de llamadas se realizan actividades diversas como: solicitud de pedidos, respuesta a inquietudes, gestión comercial y procesar reclamos y sugerencias de los clientes/compradores. Estas funciones pueden ser divididas bajo el siguiente criterio: función reactiva y proactiva[CITATION MIC18 \l 2058].

La función reactiva: Ocurre cuando un cliente llama para reclamar o quejarse y es en ese momento donde el asesor debe escuchar con mucha atención para después investigar el asunto y así lograr que el cliente esté satisfecho con lo que transmite el asesor [CITATION Sar16 \l 2058].

La función proactiva: consiste en anticiparse a los problemas y necesidades del cliente, es decir, saber qué servicios desean los clientes[CITATION Sar16 \l 2058].

La plataforma Platel utiliza como base la tecnología Freeswitch, la cual tiene implementado un módulo de centro de llamadas. Este módulo es una aplicación de cola de llamadas entrantes que se puede utilizar para las necesidades del centro de llamadas, al distribuir llamadas a los agentes utilizando varios escenarios y reglas. Sin embargo, en entrevistas realizadas a los desarrolladores se pudo constatar que la gestión de las colas de llamadas, así como los agentes son realizadas a través de comandos. Estos comandos se implementan en archivos XML y para realizar cualquier cambio es necesario modificar este archivo, lo cual resulta tedioso para los desarrolladores, además pueden cometerse errores que provoquen mal funcionamiento en la plataforma.

Sumado a lo anterior, existe un retardo en la visualización de los eventos provocando que se muestren resultados desactualizados, esto ocurre debido a que no se puede supervisar de forma concurrente todos los eventos asociados al monitoreo de agentes y colas. Otra deficiencia que presenta el centro de llamadas es que no es posible visualizar el comportamiento de las llamadas ni el desempeño de los agentes con la rapidez necesaria, ni se pueden obtener reportes históricos que ayuden a mejorar los niveles de servicio de la plataforma Platel.

1.4 Estudio de Soluciones Existentes

La dinámica de las telecomunicaciones en el mundo ha provocado un gran aumento en la implementación de Centros de Atención de Llamadas y con el desarrollo de aplicaciones que permitan medir y evaluar los eventos que tienen lugar en estos centros. A nivel internacional se destacan aplicaciones de monitoreo y gestión como: FusionPBX y QueueMetrics. A continuación, se exponen características y funcionalidades de cada una de estas.

QueueMetrics

QueueMetrics es un software que permite monitorizar y controlar el comportamiento de las colas ACD con sus agentes en una centralita Asterisk. Es un sistema de explotación de los datos de rendimiento de las colas y agentes que permite conocer el nivel de calidad que está prestando el centro de llamadas[CITATION Qua19 \l 2058].

El monitor del centro de llamadas de QueueMetrics permite realizar un seguimiento de la productividad y el tiempo de trabajo de los agentes, nóminas, objetivos de ventas, tasas de conversión, ACD, IVR y eventos de música en espera.

Genera estadísticas de campañas entrantes y salientes y supervisa los procesos en tiempo real con paneles e informes personalizables[CITATION Que23 \l 2058].

Características principales:

- ✓ Informes detallados sobre la actividad de las llamadas para colas para el tráfico entrante y saliente, filtrado por el agente y el período de tiempo.
- ✓ Permite escuchar las llamadas grabadas.
- ✓ Cuenta con un panel en tiempo real que muestra la actividad del Call Center, ver el estado del agente y el tiempo real de la actividad.
- ✓ Permite el rastreo de estados de terminación de llamada y códigos de pausa, Además puede ver las estadísticas de la actividad y duración de las llamadas teniendo un seguimiento del ACD[CITATION Qua19 \l 2058].

FusionPBX

FusionPBX es una interfaz gráfica (GUI) para Freeswitch, pero es también mucho más. A través de FusionPBX es posible añadir funcionalidades y configuraciones de una forma más cómoda y más fácil de gestionar. Viene con toda una serie de configuraciones predefinidas que permiten crear de manera rápida y sencilla conferencias audio/video, colas de espera con relativos agentes, llamadas automáticas programables, desvío, servicio, etc[CITATION Voz20 \l 2058].

FusionPBX no se usa como sistema de administración de Freeswitch ya que Platel como sistema de comunicaciones unificadas va más allá de una PBX por lo que cuenta con sistema de correo, jabber, entre otros, por tanto, se desarrolló un sistema de administración propio que cumpla con especificaciones propias de seguridad dado por el Minint y Etecsa para el despliegue de Platel.

Conclusiones de soluciones existentes

Los sistemas analizados presentan algunas de las funcionalidades necesarias para dar solución a la problemática a resolver ya que permiten monitorear y gestionar los diferentes eventos que tienen lugar en un Centro de Atención de Llamadas en tiempo real usando como núcleo Freeswitch. Sin embargo, presentan como dificultad para su uso un alto costo de adquisición y soporte, esto se debe a que son basados en software privativos en el caso de QueueMetrics. Por lo que no se pueden modificar funcionalidades existentes ni agregar nuevas, es decir, en caso de que se requiera añadir nuevas funciones de acuerdo a las características propias de la empresa no sería posible, ya que no se puede acceder a su código. Además, el pago de las patentes o licencias es un aspecto a tener en cuenta, ya que el país no se encuentra en condiciones económicas de realizar estos pagos, pues estos sistemas se ajustan a políticas capitalistas.

1.5 Metodología utilizada en la solución propuesta

Como metodología se denomina la serie de métodos y técnicas de rigor científico que se aplican sistemáticamente durante un proceso de investigación para alcanzar un resultado teóricamente válido. En este sentido, la metodología funciona como el soporte conceptual que rige la manera en que se aplican los procedimientos en una investigación.[CITATION Yan22 \l 2058]

Respetando las políticas adoptadas por el proyecto XETID del centro de Telecomunicaciones se decidió utilizar como metodología ágil la variación de Agile Unified Process (AUP por sus siglas en inglés) para la UCI, AUP-UCI. Esta metodología está orientada de manera genérica ya que para su utilización se debe adaptar a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable[CITATION Sán15 \l 2058].

Descripción de las fases

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que se

llama Ejecución y se agrega la fase de Cierre [CITATION Sán15 \l 2058]. Para una mayor comprensión se muestra la siguiente Tabla.

Fases AUP	Fases Variación AUP-UCI	Objetivos de las fases (Variación AUP-UCI)
Inicio	Inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto
✓ Elaboración ✓ Construcción ✓ Transición	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
	Cierre	En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Tabla 1.2 Historia de Usuario (Adicionar Cola)

Fuente: Elaboración Propia

AUP-UCI propone 7 disciplinas las cuales son: modelado del negocio, requisitos, análisis y diseño, implementación, pruebas internas, pruebas de aceptación y pruebas de liberación [CITATION Sán15 \l 2058]. Además, para la disciplina de requisitos la misma propone 4 escenarios, para la presente investigación, se escogió el escenario 4, ya que es un escenario ágil, los clientes estarán disponibles y acompañando al desarrollador durante toda la etapa de educación de requisitos, diseño e implementación del sistema, se tiene bien definido el negocio, el proyecto no será extenso y están bien definidos los elementos a implementar.

1.6 Tecnologías y herramientas asociadas al desarrollo del sistema

La selección de las herramientas y tecnologías, en las que se apoya el desarrollo de la aplicación web se realizó teniendo en cuenta la aplicación de ellas al contexto de trabajo del sistema, la vigencia en las comunidades de desarrollo y las facilidades funcionales que aportan al desarrollo del producto.

1.6.1 Herramientas de modelado

Para la creación de modelos y diagramas que representan el diseño y la estructura del sistema propuesta de solución se utilizarán: Visual Paradigm y UML.

Herramienta Case (Visual Paradigm for UML 17.1)

Como herramienta CASE se utiliza Visual Paradigm for UML 17.0. Visual Paradigm es una herramienta de diseño UML y herramienta CASE (del inglés Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) diseñada para ayudar al desarrollo de software. Soporta los principales estándares de la industria tales como el Lenguaje de Modelado Unificado (UML) y la notación de Modelado de Procesos de Negocio (BPMN). Ofrece un completo conjunto de herramientas de equipos de desarrollo de software necesario para la captura de requisitos, planificación de software, planificación de controles, modelado de clases y modelado de datos. Además, permite modelar todos los tipos de diagramas de clases, código inverso y generar código desde los diagramas. [CITATION Yan22 \l 2058]

Proporciona una plataforma ampliable para que los desarrolladores puedan agregar funciones al mismo, ellos pueden hacer referencia a los plugin (componentes) de guiar el desarrollo, a construir sus propios plugin para leer, actualizar, recuperar y eliminar los diagramas y los elementos del modelo.

UML 2.0

Se utiliza UML en su versión 2.0 como lenguaje de modelado, ya que el mismo permite al equipo de desarrollo visualizar, especificar, construir y documentar los artefactos del sistema. Esto proporciona una forma estándar de escribir los planos del mismo y cubrir tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables.

Por lo antes planteado fue seleccionada esta herramienta para llevar a cabo el proceso de modelado de la solución propuesta en esta investigación.

1.6.2 Gestor de bases de datos y Base de datos

Para la administración y gestión de base de datos se seleccionaron: PgAdmin y PostgreSQL, pues brindan alto rendimiento, fácil integración, escalabilidad y seguridad de los datos.

Base de datos PostgreSQL 15.3

PostgreSQL es un sistema de código abierto de administración de bases de datos del tipo relacional, aunque también es posible ejecutar consultas que sean no relacionales. Dos detalles a destacar de PostgreSQL es que posee data types (tipos de datos) avanzados y permite ejecutar optimizaciones de rendimiento avanzadas, que son características que por lo general solo se ven en sistemas de bases de datos comerciales, como por ejemplo SQL Server de Microsoft u Oracle de la compañía homónima.[CITATION Yan22 \l 2058]

Se decide usar esta herramienta ya que PostgreSQL es compatible con numerosas plataformas de alta disponibilidad y esta solución permite soportar grandes cargas de trabajo.

PgAdmin4

PgAdmin 4 es la principal herramienta de gestión de código abierto para Postgres. PgAdmin está diseñado para monitorear y administrar múltiples servidores de bases de datos PostgreSQL, tanto locales como remotos, a través de una sola interfaz gráfica que permite la fácil creación y administración de objetos de bases de datos, así como una serie de otras herramientas para administrar sus bases de datos. [CITATION Yan22 \l 2058]

1.6.3 Entorno de Desarrollo Integrado (IDE)

Una de las herramientas que desempeña un papel importante en el desarrollo de soluciones informáticas son los Entornos de Desarrollo Integrado (IDE). Estos ofrecen facilidades al equipo de desarrollo cuando se implementan las aplicaciones debido a que permite la identificación de errores comunes que se comenten a diario. Para el desarrollo de la propuesta de solución, se seleccionó Visual Studio Code, pues este proporciona una

interfaz gráfica de usuario que incluye herramientas y funciones para la escritura, depuración, compilación y pruebas de código. Al proporcionar un conjunto completo de herramientas en una sola plataforma, permite trabajar de manera más rápida reduciendo el tiempo y el esfuerzo requeridos para desarrollar y mantener software de calidad

Visual Studio Code versión 1.78

Visual Estudio Code es una versión reducida del entorno de desarrollo oficial de Microsoft centrado exclusivamente en el editor de código. Es multiplataforma y soporta la sintaxis de una gran cantidad de lenguajes de programación. La herramienta proporciona soporte y asistencia a lenguajes de diversos ámbitos: HTML, CSS, JavaScript, diversas variantes de C, Java, SQL, PHP, Ruby, Visual Basic o JSON entre otros, soportando resultado, snippets y autocompletado.[CITATION Yan22 \l 2058]

A pesar que para el desarrollo de la aplicación se pudo seleccionar cualquier otro entorno, la selección de Visual Studio 1.78 está ligada al hecho de la acumulada experiencia de trabajo con este IDE además de la avanzada programación que se tenía de ante mano (reutilización de código) y asegura al usuario un entorno de trabajo amistoso y fácil de asimilar

1.6.4 Tecnologías

Para el desarrollo de la propuesta de solución se seleccionaron las tecnologías mencionadas a continuación porque proporcionan herramientas y características que facilitan el proceso de construcción de aplicaciones web modernas, rápidas y seguras.

Django versión 4.2.2

Django es un marco de trabajo (framework) para el desarrollo de aplicaciones web usando Python. Django considera algunas funcionalidades listas para facilitar el desarrollo de aplicaciones web. Como resultado, no es necesario escribir todo el código ni usar tiempo para buscar errores de código en el framework. Es decir, mediante Django, el desarrollo de sistemas de información web puede ser rápido, seguro, escalable y también fácil de mantener.

Django es un framework web de alto nivel de código abierto, que respeta el patrón de diseño conocido como modelo–vista–controlador (MVC). Permite el desarrollo rápido de sitios web seguros y mantenibles. La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes; proporciona una serie de características que facilitan el desarrollo rápido de páginas orientadas a contenidos.

Este framework proporciona una aplicación incorporada para administrar los contenidos, que puede incluirse como parte de cualquier página hecha con Django y que puede administrar varias páginas a partir de una misma instalación; la aplicación administrativa permite la creación, actualización y eliminación de objetos de contenido, llevando un registro de todas las acciones realizadas sobre cada uno, y proporciona una interfaz para administrar los usuarios y los grupos de usuarios[CITATION Yas22 \l 1033].

Bootstrap versión 5.3

Bootstrap es un framework basado en HTML y CSS, creado por Twitter y liberado en 2012. Desde entonces ha ganado muchos adeptos, hasta el nivel de contar con la mayor comunidad de Github del mundo. Este framework ayuda a agilizar la creación de la interfaz de una aplicación web. Su uso permite que el sitio web sea adaptable a la pantalla del dispositivo con el que se accede, ya sea un ordenador, tablet, smartphone, televisión. Esto significa que se tendrá una web sensible o adaptativa[CITATION boo22 \l 1033].

La elección de la tecnología fue escogida en conjunto con los desarrolladores del centro de desarrollo.

1.6.5 Lenguajes de programación

Python versión 3.11.3

Se utiliza Python como lenguaje de programación ya que es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo. A diferencia de otros lenguajes como Java o .NET, se trata de un lenguaje interpretado, es decir, que no es necesario compilarlo para ejecutar las aplicaciones escritas en Python, sino que se ejecutan directamente por el ordenador utilizando un programa denominado interpretador, por lo que no es necesario “traducirlo” a lenguaje máquina. (Universidades, 2021)

JavaScript

Es un lenguaje de programación multipropósito que se utiliza principalmente para la creación de aplicaciones web del lado del cliente, pero también puede ser utilizado en el lado del servidor, aplicaciones de escritorio y aplicaciones móviles. Su sintaxis se asemeja a la de otros lenguajes de programación como C, C++ y Java, pero también tiene características únicas, como su sistema de tipado dinámico y su enfoque en la programación basada en eventos[CITATION Jav23 \l 1033].

Otra característica clave de JavaScript es su capacidad de trabajar con eventos, lo que permite que el código responda a las acciones del usuario, como hacer clic en un botón o mover el mouse sobre un elemento. Esto permite crear experiencias de usuario más ricas y personalizadas[CITATION Jav23 \l 1033]

En resumen, las herramientas y tecnologías fueron seleccionadas debido a su amplia funcionalidad y compatibilidad con el entorno de desarrollo donde se está trabajando. Estas elecciones se basaron en la popularidad y la comunidad activa que rodea a estas tecnologías, lo que facilitó la colaboración con otros desarrolladores y aceleró el proceso de desarrollo. Asimismo, estas herramientas proporcionan las funcionalidades necesarias para crear una interfaz de usuario moderna y agregan interactividad y dinamismo a las aplicaciones web. En conjunto, estas elecciones permitieron aprovechar el conocimiento y las experiencias existentes en el entorno de trabajo.

A pesar de lo anteriormente expuesto, se trabajó con estos lenguajes por la vasta experiencia, conocimiento y dominio de los desarrolladores.

1.7 Conclusiones parciales

A lo largo de este capítulo se han abordado los diferentes elementos teóricos sobre los cuales se sustenta esta investigación, por lo que se puede concluir:

1. La descripción de los principales conceptos permitió adquirir una mayor comprensión de los temas relacionados con el objeto de estudio de la presente investigación.
2. Se llevó a cabo un estudio de soluciones similares lo cual permitió tener una mejor comprensión del problema y posibles métodos de solución al mismo.
3. Se propusieron las principales herramientas y tecnologías para desarrollar una aplicación que permita monitorear las colas y agentes de un Centro de Atención de Llamadas. Se utilizará el IDE Visual Studio 1.78 para la programación de la aplicación, Visual Paradigm para su modelado y AUP-UCI como metodología de desarrollo.

Capítulo 2: Propuesta de Solución

En el presente capítulo se describe la aplicación que se desea implementar. Se lleva a cabo el levantamiento de requisitos, los cuales se especifican como funcionales y no funcionales. Se encapsulan los requisitos mediante historias de usuarios como se define en el escenario 4 de la metodología AUP-UCI. Además, se explican los patrones de diseño para la implementación de la aplicación. Se define los patrones de diseño del software y por último, se define el modelo de datos de la base de datos que se empleará.

2.1 Análisis de la propuesta de solución para gestionar y monitorear los Agentes y Colas del centro de llamadas de la plataforma Platel.

Se propone el análisis y diseño de una aplicación web que permita gestionar las colas, y agentes de los Centros de Atención de Llamadas, así como el monitoreo de los mismos en la plataforma Platel. Mediante una aplicación web se realizará la interacción con el módulo de un centro de llamadas de Platel a través de las API del módulo (mod_callcenter) de Freeswitch. De esta forma se facilita que la aplicación sea capaz de interactuar con Freeswitch de una manera más sencilla.

El sistema está compuesto por dos subsistemas.

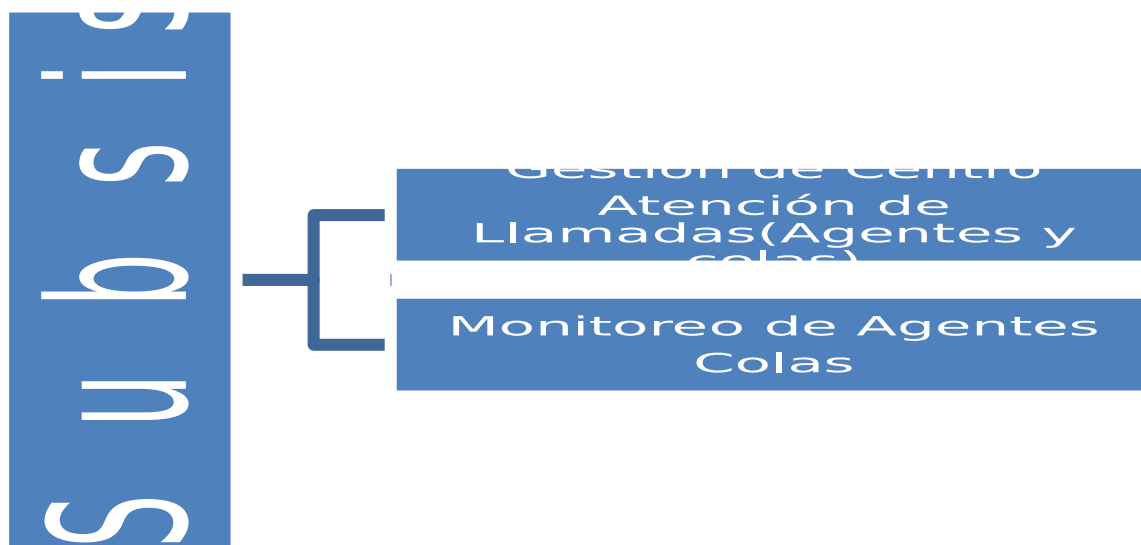


Figura 2.1. Subsistemas de Monitoreo, Supervisión y Gestión

Fuente: Elaboración propia

El primero de los subsistemas es el encargado de la gestión de los Centros de Atención de Llamadas, ya sea insertarlos, modificarlos o eliminarlos. El segundo subsistema tiene como objetivo supervisar los eventos ocurridos en el Centro de Atención de Llamadas con el fin de examinar el estado actual y el desempeño de los agentes. Además es el encargado de

monitorear las llamadas asociadas a los Centro de Atención de Llamadas en cuanto a intervención y estado de las mismas.

2.1.1 Descripción de la propuesta de solución del sistema

El administrador al acceder al subsistema Gestión de Centro de Atención de Llamadas puede realizar las funcionalidades de insertar, modificar los agentes y las colas. Para insertar un nuevo agente el usuario debe entrar datos como: nombre, tipo, contacto, estado, máximo_sin_respuesta, tiempo_conclusión, tiempo_retraso_rechazo, tiempo_retraso_ocupado, tiempo_retraso_sin_respuesta; si se desea se pueden modificar los datos entrados de agente previamente insertado. En caso de optar la opción de eliminar agente debe seleccionar el deseado a dar de baja. También puede realizar la adición de una cola al entrar datos como: nombre, estrategia, puntuación_base_tiempo, curriculum_abandonado_permitido, tiempo_max_espera, regla_nivel_sin_agente, aplicar_regla_nivel, regla_nivel_espera_segundos, multiplicar_nivel, descarte_abandonado, tiempo_max_espera_sin_agente, si se desea se pueden modificar los datos entrados de cola previamente insertado. En caso de optar la opción de eliminar cola debe seleccionar el deseado a dar de baja. Por último, se debe asociar a las colas creadas los agentes que van a atender las llamadas que llegan a las mismas.

Para poder supervisar los eventos ocurridos en el Centro de Atención de Llamadas el supervisor debe acceder al subsistema Monitoreo, donde tiene la posibilidad de supervisar las actividades desarrolladas por los agentes, el estado de las colas de llamadas. En la supervisión de agentes se visualizan diferentes datos de los mismos, como son: nombre del agente, tiempo conectado, tiempo hablando, tiempo ocupado, las llamadas atendidas y el estado actual, que puede ser Disponible, Hablando, Ocupado o Desconectado. Los datos obtenidos para la supervisión de la cola de llamadas son: nombre de la cola, cantidad de llamadas que están siendo atendidas, en espera y abandonadas.

2.3 Definición de los requisitos de la Aplicación

La ingeniería de requisitos es la fase de un proyecto de software donde se definen las propiedades y la estructura del mismo; y que a la vez comprende el desarrollo y gestión de requisitos[CITATION Tor16 \l 3082].

2.3.1 Especificación de los requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos

funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer [CITATION Som06 \l 3082].

De cada uno de los requisitos funcionales se especifica su descripción, prioridad (P) y complejidad(C).

En el proceso de desarrollo de la aplicación web se definió el nivel de prioridad para el negocio de cada uno de los requisitos obtenidos en la entrevista con el cliente. Los niveles se dividen en:

Alta: Funcionalidades que son sumamente primordiales en la aplicación, que por su función no deberían faltar en el editor, es decir, la aplicación depende de estas para su correcto funcionamiento.

Media: Funcionalidades que no son muy primordiales, pero otras dependen de estas para poder ejecutarse.

Baja: Funcionalidades que su existencia en la aplicación no es significativa, es decir, la aplicación puede funcionar correctamente sin estas.

En la siguiente tabla 1.2 se realizó la descripción de cada requisito funcional con la prioridad y complejidad de cada uno.

N°	Nombre	Descripción	P	C
RF 1	Registrar Usuario	El sistema permite que un usuario pueda crearse una cuenta permanente para acceder al sistema.	Baja	Alta
RF 2	Autenticar Usuario	El sistema permite a los usuarios acceder al sistema mediante usuario y contraseña.	Alta	Alta
RF 3	Cerrar Sesión	El sistema permite al usuario cerrar su sesión para salir del sistema.	Baja	Baja
RF 4	Listar Usuario	El sistema permite hacer un listado de todos los usuarios registrados en él.	Alta	Baja
RF 5	Modificar Usuario	Luego de autenticado un usuario, el sistema permite que este pueda modificar sus datos de usuario.	Alta	Media

RF 6	Adicionar Usuario	Los administradores podrán crear usuarios dentro del sistema.	Alta	Media
RF 7	Eliminar Usuario	Permite al usuario con los permisos suficientes eliminar a otro usuario registrado en el sistema.	Alta	Media
RF 8	Buscar Usuario por filtros	El sistema permite buscar a un usuario del sistema por: Nombre,	Media	Baja
RF 9	Listar Agente	El sistema permite hacer un listado de todos los agentes registrados en él.	Alta	Baja
RF 10	Adicionar Agente	Los administradores u otros usuarios con los permisos asignados podrán crear agentes dentro del sistema.	Alta	Media
RF 11	Modificar Agente	Luego de autenticado un agente, el sistema permite que este pueda modificar sus datos y el de otros según los permisos que tenga asignado.	Alta	Media
RF 12	Eliminar Agente	Permite al usuario con los permisos suficientes eliminar a un agente registrado en el sistema.	Alta	Media
RF 13	Buscar Agente por filtros	El sistema permite buscar a un agente del sistema por: Nombre,	Media	Baja
RF 14	Listar Cola	El sistema permite hacer un listado de todas las colas del sistema	Alta	Baja
RF 15	Adicionar Cola	El sistema permite al administrador adicionar una cola	Alta	Media
RF 16	Modificar Cola	El sistema permite al administrador modificar los datos de una cola	Alta	Media
RF 17	Eliminar Cola	Permite al usuario con los permisos suficientes eliminar una cola	Alta	Media

RF 18	Buscar Cola por filtros	El sistema permite buscar una cola del sistema por: Nombre	Media	Baja
RF 19	Listar centro de llamada	El sistema permite listar los agentes asociados a la cola que se encuentra, así como el nivel y posición que este ocupa en la misma.	Alta	Baja
RF 20	Modificar centro de llamada	El sistema permite modificar el nivel y la posición de un agente en una cola determinada.	Alta	Media
RF 21	Eliminar centro de llamada	El sistema permite eliminar un agente de determinada cola.	Alta	Media
RF 22	Adicionar centro de llamada	El sistema permite asociar un agente a una cola y asignarle su posición y nivel.	Alta	Media
RF 23	Mostrar Cantidad de agentes	El sistema permite ver la cantidad de agentes que están creados.	Alta	Media
RF 24	Mostrar la Cantidad de agentes atendiendo llamadas	El sistema permite ver la cantidad de agentes atendiendo llamadas.	Alta	Alta
RF 25	Mostrar Cantidad de agentes registrados	El sistema permite ver la cantidad de agentes que están registrados para recibir llamadas.	Alta	Alta
RF 26	Mostrar Cantidad de colas en el sistema	El sistema permite ver cuantas llamadas están esperando ser atendidas en una cola determinada.	Alta	Alta
RF 27	Monitoreo de agentes	El sistema le permite al supervisor monitorear los agentes en tiempo real para saber su estado, cantidad de llamadas realizadas, el tiempo de conversación, el nivel de la cola, el agente y su status.	Alta	Alta
RF 28	Monitoreo de colas	El sistema le permite al supervisor monitorear las colas en tiempo real para saber su estado, nombre, su posición, el identificador y el tiempo que espera a ser atendida.	Alta	Alta

Tabla 2.1 Requisitos Funcionales

Fuente: Elaboración Propia

2.3.2 Especificación de requisitos no funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema [CITATION Som06 \l 3082].

En la solución propuesta que se integrará a la PBX Platel, los requisitos mínimos de hardware deben ser los mismos que tiene Platel para el despliegue en las entidades u organismos que lo posean, los cuales son:

Hardware:

✓ Software

- **Para el cliente:**
 - Navegador web: Mozilla Firefox v3.6 o superior.
- **Para el servidor:**
 - Debe estar configurado con las extensiones PDO, PDO_pgsql y pgsql.
 - Gestor de base de datos: PostgreSQL v9.1.
 - Sistema operativo: Linux Ubuntu v12.4.
 - Se debe tener publicada la información del CDR que genera la PBX Yate.

✓ Hardware

- **Para las PC clientes:**
 - RAM: 128 MB o superior.
 - Disco duro: 10 GB o superior.
- **Para los servidores:**
 - Procesador: Doble Micro a 3.0 GHz o superior.
 - RAM: 2 GB o superior.
 - Disco duro: 20 GB o superior.
 - Procesador: Doble Micro a 90 GHz o superior.
 - RAM: 2 GB o superior.

- Disco duro: 500 GB o superior.
- UPS: 1.
- Tarjeta de Red: 1 GB.

2.3.3 Descripción de requisitos mediante Historias de Usuarios.

Las Historias de Usuarios son una herramienta que agiliza la administración de requisitos, reduciendo la cantidad de documentos formales y tiempo necesario. Se usan, en el contexto de la ingeniería de requisitos ágiles, como una herramienta de comunicación que combina las fortalezas de ambos medios: escrito y verbal.

En lo adelante, se hace una descripción de lo que se espera una vez implementado el requisito y el beneficio que aporta al usuario.

<i>Historia de Usuario #2: Autenticar usuario</i>	
Número: 2	Nombre del requisito: Autenticar usuario
Programador: Glenda Grant Varona	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 2 días
Riesgo en Desarrollo: Alta	Tiempo Real: 2 días
Descripción: Una vez registrado el usuario, el sistema le permite iniciar sesión y acceder a las funcionalidades que le fueron asignadas según el rol que desempeñe el mismo.	
Observaciones: Acciones para lograr el objetivo (Precondiciones y Datos): • Ingresar los datos de Usuario y Contraseña. Comportamientos Válidos y No Válidos (Flujo Central y Alterno): • Usuario: No admite ni caracteres extraños, ni numéricos, ni especiales. • Contraseña: Campo que admite un mínimo de 8 caracteres, combinados entre mayúscula, minúscula y caracteres especiales Flujo de la Acción a Realizar: • El sistema debe permitir la autenticación • Cuando el usuario toca el botón iniciar sesión ubicado en la parte superior derecha de la aplicación, se muestra una interfaz donde debe ingresar sus datos (usuario y contraseña), luego selecciona la opción iniciar sesión y si son correctos los datos inicia sesión en la aplicación.	

- Si los datos son incorrectos sale un mensaje que dice usuario o contraseña incorrecta y da la posibilidad de realizar esa acción nuevamente.

Prototipo de Interfaz:

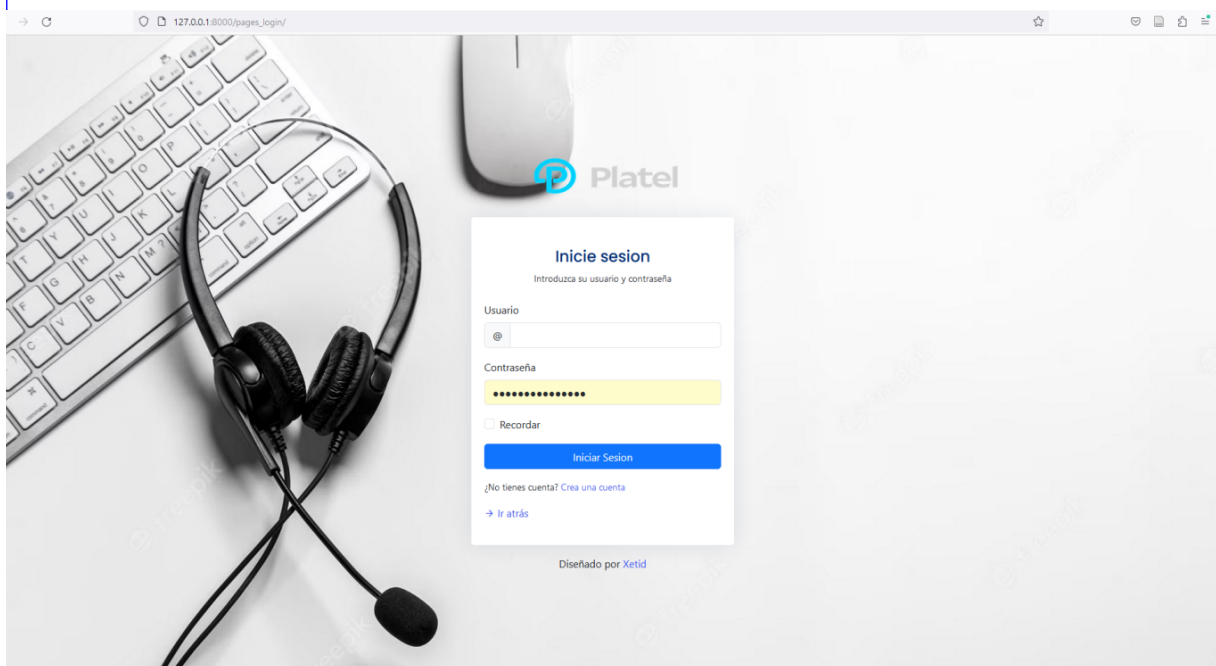


Tabla 2.2 Historia de Usuario (Autenticar Usuario)

Fuente: Elaboración Propia

Historia de Usuario #2: Adicionar Agente

Número: 2	Nombre del requisito: Adicionar Agente
Programador: Glenda Grant Varona	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 4 días
Riesgo en Desarrollo: Alta	Tiempo Real: 4 días
Descripción: Esta funcionalidad debe permitir al administrador añadir los agentes al sistema.	
Observaciones: Se deben manejar todas las restricciones requeridas por el cliente.	
Prototipo de Interfaz:	

ADICIONAR AGENTE

Introduzca su nombre

Introduzca el tipo de agente

Seleccione el estado del agente predeterminado

Introduzca el tiempo de conclusion

Introduzca el tiempo de retraso ocupado

user/name

Ingrese el tiempo maximo sin respuesta

Ingrese el tiempo de retraso de rechazo

Introduzca el tiempo de retraso sin respuesta

Adicionar Cancelar

Tabla 2.3 Historia de Usuario (Adicionar Agente)

Fuente: Elaboración Propia

Historia de Usuario #2: Adicionar Cola	
Número: 2	Nombre del requisito: Adicionar Cola
Programador: Glenda Grant Varona	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en Desarrollo: Alta	Tiempo Real: 5 días
Descripción: Esta funcionalidad debe permitir al administrador añadir las colas al sistema.	
Observaciones: Se deben manejar todas las restricciones requeridas por el cliente.	
Prototipo de Interfaz: ADICIONAR COLA Introduzca el nombre Introduzca la estrategia Introduzca el curriculum abandonado permitido Ingrese regla de nivel sin agente Ingrese la regla de nivel de espera en segundos Ingrese el descarte abandonado permitido Introduzca la puntuacion base tiempo Establezca el tiempo maximo de espera Aplicar la regla de nivel Ingrese la regla para multiplicar nivel Ingrese el tiempo maximo de espera sin agente Enviar Cancelar	

Tabla 2.4 Historia de Usuario (Adicionar Cola)

Fuente: Elaboración Propia

Diseño de la solución.

Una vez que se han identificado las historias de usuarios y se ha realizado la planificación de las iteraciones, los diagramas de clases de diseño pueden ser creados para definir la estructura del sistema y las clases que serán necesarias para implementar las funcionalidades solicitadas en las historias de usuario.

Los diagramas de clases de diseño establecen una conexión entre los requisitos funcionales del sistema y su implementación técnica. Los diagramas de clases de diseño proporcionan una representación visual que facilita la comprensión de cómo se organizarán las clases y cómo interactuarán entre sí para cumplir con las funcionalidades requeridas.

En el siguiente Diagrama de Clases del Diseño se muestra el funcionamiento de los requisitos relacionados con los Agentes, en este caso son: adicionar Agente, modificar Agente, eliminar Agente. La Client Page es la parte del frontend de la aplicación que va a enviar una petición a la Server page con los datos a través del formulario necesarios para incluir el nuevo Agente. La petición es recibida por la Server Page y esta busca la vista que se encarga de gestionar esta petición.

En la presente investigación se creó un diagrama de clases de diseño con estereotipos web, correspondiente a uno de los principales requisitos descritos en el epígrafe 2.3.1.

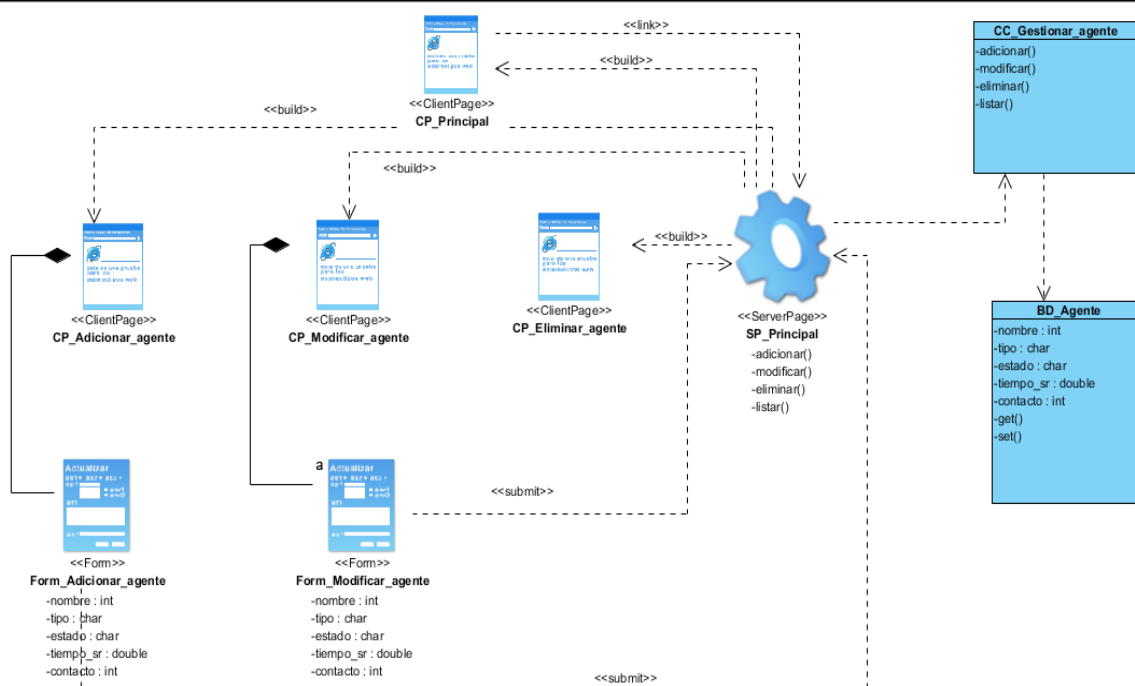


Figura 2.2. Diagrama de clases del diseño de la Historia de Usuario Adicionar Agente

Fuente: Elaboración Propia

En este caso, desde la CP_principal se puede acceder a la CP_Añadir_agente mediante el SP_Principal, al solicitar esta funcionalidad se genera el formulario Añadir Agente haciendo uso de la clase Agente, los datos y las solicitudes son manejados a través de la clase controladora CC_Gestionar_agente.

Esta misma lógica siguen el resto de las clases y funcionalidades implementadas en el sistema.

2.4. Arquitectura de software

La arquitectura de un software o sistema de cómputo, es la estructura que comprende a los componentes del mismo, sus propiedades externas visibles y las relaciones entre ellos. Es una representación que permite: analizar la efectividad del diseño para cumplir los requerimientos establecidos, considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil y reducir los riesgos asociados con la construcción del software[CITATION Pre10 \l 3082].

La arquitectura de software de la aplicación se basa en Django, un framework que emplea el patrón de diseño Model-Views-Template (MVT). Esta estructura es similar al tradicional Modelo-Vista-Controlador (MVC) y condiciona la organización del software de la aplicación.

2.4.1 Patrón Arquitectónico

El patrón arquitectónico propone una solución arquitectónica que sirve como base para el diseño de la arquitectura e impone la transformación del diseño la misma. Además, describe la manera en la que el software manejará ciertos aspectos de su funcionalidad en el nivel de la infraestructura (Pressman 2010).

En el patrón de arquitectónico utilizado por Django, se separa la lógica de la aplicación en tres componentes interconectados para facilitar la gestión y el mantenimiento del código. A continuación, se detallan estos componentes[CITATION Dis23 \l 2058]:

1. Model (Modelo): Representa la estructura de la base de datos y define los tipos de datos y relaciones entre ellos. Django proporciona una interfaz de base de datos, conocida como Object-Relational Mapping (ORM), que permite interactuar con la base de datos utilizando el lenguaje Python en lugar de SQL. Esto simplifica el manejo de la base de datos al poder acceder a las tablas como si se tratase de objetos Python. En la aplicación, los modelos se definen en el archivo "models.py"[CITATION Dis23 \l 2058].

```

from venv import create
from django.db import models
#from django.contrib.auth.models import User

class Agente(models.Model):
    nombre = models.CharField(max_length=50)
    tipo = models.CharField(max_length=50)

```

Figura 2.3. Capa de Modelo

Fuente: Elaboración Propia

2. Views (Vista): Contiene la lógica de negocio y maneja las peticiones y respuestas HTTP. Interactúa con el modelo para obtener los datos y los pasa al template correspondiente. Actúa como un intermediario entre los modelos y las templates. En la aplicación, el archivo de vista se denomina "views.py"[CITATION Dis23 \l 2058].

```

def registrar_usuario (request):
    if request.method == 'GET':
        return render(request, 'registrar_usuario.html', {'form': UserCreationForm})
    else:
        #Register user
        if request.POST['password1'] == request.POST['password2']:
            try:
                user = User.objects.create_user(username=request.POST['username'], password=request.POST['password1'], name=request.POST['name'])
                user.save()
                login(request, user)
                return redirect('index_general')
            except:
                return render(request, 'registrar_usuario.html', {'form': UserCreationForm, "error": 'Usuario ya existe'}, )
        return render(request, 'registrar_usuario.html', {'form': UserCreationForm, "error": 'Contraseñas no coinciden'})

def index_general(request):
    return render(request, 'index_general.html')

```

Figura 2.4. Capa de Views

Fuente: Elaboración Propia

3. Template (Plantilla): Representa la presentación de los datos en el modelo MVC. Los templates son archivos HTML en los que se pueden manipular los datos proporcionados por la vista. Se encargan del aspecto visual de la aplicación (del frontend) y se ubican en la carpeta "templates"[CITATION Dis23 \l 2058].

```

> modificar_agente.html
> modificar_cola.html
> modificar_usuario.html
> pages_login.html
> perfil.html

```

Figura 2.5. Capa de templates

Fuente: Elaboración Propia

De esta manera, la arquitectura Modelo-Vista-Plantilla de Django organiza claramente los componentes y facilita el desarrollo y mantenimiento de la aplicación web. La arquitectura que sigue Django queda representada en la figura.



Figura 2.6. Modelo-Vista-Plantilla [CITATION Mon13 \l 2058]

2.4.2 Patrones de diseño

Los patrones de diseño son una solución general, reutilizable y aplicable a diferentes problemas de diseño de software. Se trata de plantillas que identifican problemas en el sistema y proporcionan soluciones apropiadas a problemas generales a los que se han enfrentado los desarrolladores durante un largo periodo de tiempo, a través de prueba y error [CITATION Can20 \l 3082].

2.4.1.1 Patrones Generales de Software para la Asignación de Responsabilidades (GRASP)

Los patrones generales de software para asignar responsabilidades del inglés (General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones [CITATION Ort21 \l 2058]. Entre los patrones GRASP a utilizar se encuentran:

Controlador: Es un patrón que sirve como intermediario entre una interfaz y el algoritmo que la implementa, de tal forma que recibe los datos del usuario y los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocio debe estar separada de la capa de presentación, para aumentar la reutilización de código y a la vez tener un mayor control [CITATION Ort21 \l 2058].

```
def index_general(request):
    Agente = Monitoreo_Agente.objects.all()
    Cola = Monitoreo_Cola.objects.all()
    return render(request, 'index_general.html', {'Agente':Agente, "Cola":Cola,})
```

Figura 2.7. Ejemplo de patrón controlador

Fuente: Elaboración Propia

Experto: Con este patrón se mantiene el encapsulamiento de la información en una clase a la cual le será asignada toda la responsabilidad de creación de objetos e implementación de métodos acorde a la lógica del negocio que le corresponda.

```
class Agente(models.Model):
    nombre = models.IntegerField()
    tipo = models.CharField(max_length=5000)
    contacto = models.CharField(max_length=5000)
    estado = models.TextField(max_length=5000, choices=agent_estado)
    max_sin_respuesta = models.IntegerField()
    tiempo_conclusion = models.IntegerField()
```

Figura 2.8. Ejemplo de patrón experto

Fuente: Elaboración Propia

Alta cohesión: la cohesión es una medida de cuán relacionada y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme[CITATION Ort21 \l 2058].

Bajo acoplamiento: este patrón es un principio que asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión[CITATION Ort21 \l 2058]. Las URLconfs de Django son un claro ejemplo de este principio en la práctica. En una aplicación de Django, la definición de la URL y la función de vista que se llamará están débilmente acopladas.

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('base/', views.base, name='base'),
    path('cola/', views.colaTable, name='cola'),
    path('datoscola/', views.listarCola, name='datosCola'),
    path('agente/', views.agenteTable, name='agente'),
    path('datosagente/', views.listarAgente, name='datosAgente'),
    path('error_404/', views.error_404, name='error_404'),
    path('accounts/login/', views.pages_login, name='pages_login'),
    path('perfil/', views.perfil, name='perfil'),
    path('pages_login/', views.cerrar_sesion, name='logout'),
```

Figura 2.9. Ejemplo de patrón bajo acoplamiento

Fuente: Elaboración Propia

Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. Este patrón se usa con el objetivo de que la clase que maneja la información de los nuevos objetos a crear sea la responsable de crearlos[CITATION Ort21 \l 2058].

2.4.1.2 Patrones de diseño Gof ((por sus siglas en inglés Gang Of Four)

Los patrones Gof describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos y se agrupan en tres grandes categorías:

creacionales, estructurales y los de comportamiento[CITATION Pre10 \l 2058]. A continuación, se describe el patrón Gof utilizado en la solución propuesta:

- ✓ Decorador: Patrón estructural que extiende la funcionalidad de un objeto dinámicamente de manera tal que es transparente a sus clientes, utiliza una instancia de una subclase de la clase original que delega las operaciones al objeto original. Este patrón se evidencia en la clase Login_require (decorador que trae Django por defecto, para acceder a una clase).

```
@login_required
def index_general(request):
    Agente = Monitoreo_Agente.objects.all()
    Cola = Monitoreo_Cola.objects.all()
    return render(request, 'index_general.html', {'Agente':Agente, "Cola":Cola,})
```

Figura 2.10. Ejemplo de Patrón Gof (Decorador)

Fuente: Elaboración Propia

2.5 Modelo de Datos

El modelo de datos se utiliza para describir la estructura lógica y física de la información persistente gestionada por el sistema. Se utiliza para definir la correlación entre las clases de diseño persistentes y las estructuras de datos persistentes, y para definir las estructuras de datos persistentes[CITATION Dar22 \l 2058].

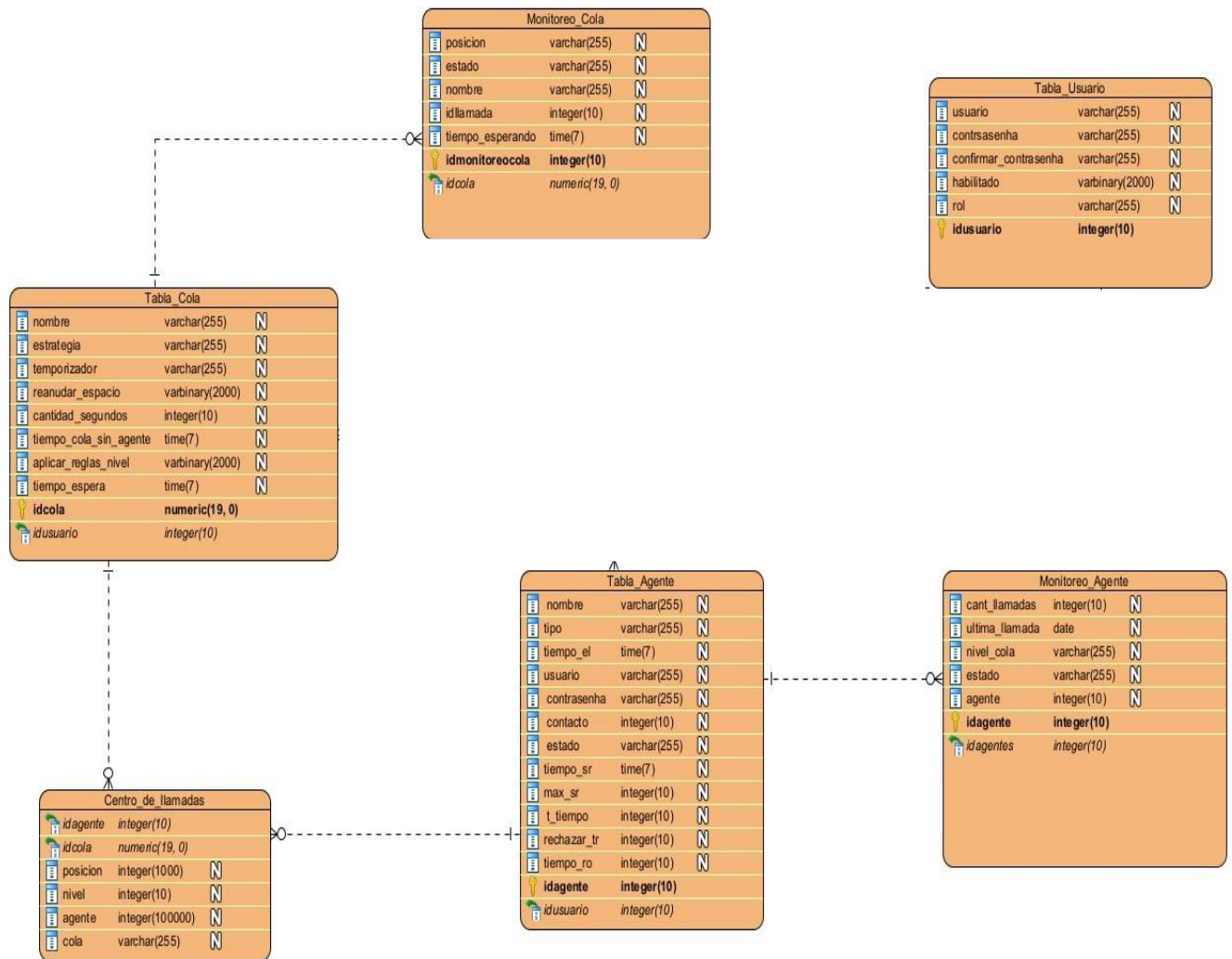


Figura 2.11. Modelo de Datos

Fuente: Elaboración Propia

En el diagrama de modelo de datos se ven representadas la relación entre las clases donde se describirá la relación de las tablas representadas en la figura, brindando un análisis detallado de sus estructuras, interconexiones y su papel fundamental en el almacenamiento y organización de los datos en un modelo de base de datos.

Tabla_agente: Esta tabla contiene los agentes del sistema con los atributos: nombre, tipo, estado, contacto, entre otros. La cual se relaciona con la tabla Monitoreo_agente con una relación de uno a uno, lo que significa que un agente puede tener un registro de monitoreo asociado a él.

TablaCola: Esta tabla contiene las colas del sistema con los atributos: nombre, estrategia, tiempo-espera, temporizador, entre otros. La cual se relaciona con la tabla MonitoreoCola con una relación de uno a uno, lo que significa que una cola puede tener un registro de monitoreo asociado a él.

Monitoreo_cola: Esta tabla almacena la información de monitoreo específica para cada cola lo cual incluye los siguientes atributos: posición, id llamada, estado, nombre, entre otros.

Monitoreo_agente: Esta tabla almacena la información de monitoreo específica para cada agente lo cual incluye los siguientes atributos: cantidad de llamadas, nivel d cola, estado, agente, entre otros.

Centro_llamadas: Esta tabla actúa como una tabla de unión que relaciona las tablas Tabla_agente y Tabla-cola en el centro de llamadas. La relación entre el centro de llamada y las otras dos tablas es de uno a muchos, lo que implica que un centro de llamadas puede tener múltiples agentes y colas asociadas a él.

2.6 Conclusiones parciales

En el presente capítulo se abordaron los temas referentes al desarrollo de la aplicación, por lo que se puede concluir que:

1. El análisis de la propuesta de solución permitió con el levantamiento de los requisitos del sistema, determinar las funcionalidades básicas a desarrollar durante el proceso; definiendo 28 HU.
2. El análisis y diseño permitió seleccionar el patrón arquitectónico Modelo Vista Template y se explica cómo se pusieron de manifiesto los patrones GRASP y GOF en la implementación de la solución.
3. Al emplear el lenguaje UML para modelar los artefactos propuestos por la metodología de desarrollo seleccionada durante el capítulo 1 se alcanzó una mejor comprensión sobre cómo debe comportarse la aplicación web.

CAPÍTULO 3: PRUEBAS O VALIDACIÓN

En el presente capítulo se muestran los resultados alcanzados en la implementación del sistema, así como algunas interfaces de la aplicación. Se evidencia la organización del sistema mediante el diagrama de componentes el cual representa la vista estática del mismo, y mediante el diagrama de despliegue se representa cómo se distribuirán los servicios que brindará el sistema. Por último, se describirán las pruebas realizadas al sistema y los resultados obtenidos una vez aplicadas las mismas; además de la validación de la investigación.

3.1 Diagrama de Componentes

Los diagramas de componentes constituyen en esta disciplina el modelo de implementación, al describir los componentes a construir, su organización y dependencias. Un componente es una parte física y reemplazable de un sistema que se conforma por un conjunto de interfaces y proporciona la realización de dicho conjunto. Estos diagramas se usan para modelar los elementos físicos que pueden hallarse en un nodo por lo que empaquetan elementos como clases, colaboraciones e interfaces. A continuación, se muestran los diagramas de componentes [CITATION Ari19 \l 2058].

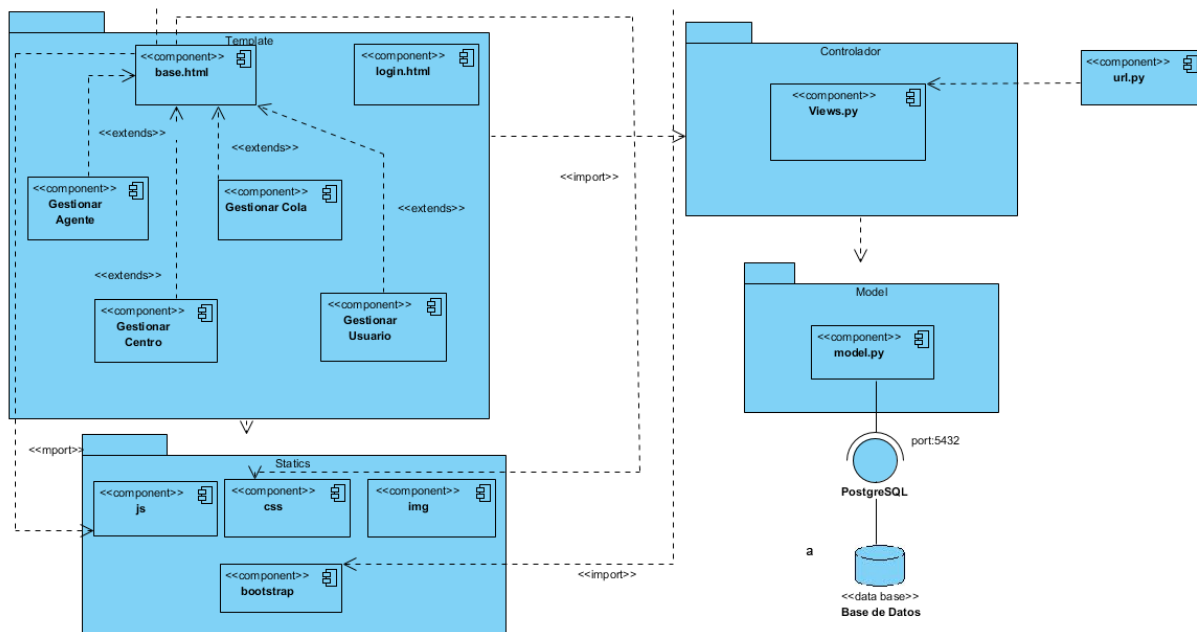


Figura 3.1. Diagrama de componentes

Fuente: Elaboración Propia

Descripción del diagrama de componentes del sistema

En el diagrama de componentes mostrado anteriormente está basado en el patrón arquitectónico MTV (Modelo-Vista-Template), el cual se basa en la separación de responsabilidades en el desarrollo web. El paquete Template se encarga de la presentación

visual de la aplicación, el paquete Controlador maneja la lógica de negocio y la interacción con el cliente, y el paquete modelo gestiona la capa de datos. El componente urls.py define las rutas de acceso a las vistas, y el paquete Static almacena los archivos estáticos, incluyendo framework Bootstrap para mejorar el diseño y la usabilidad de las páginas web.

Una vez especificado el diagrama de componentes, se define el diagrama de despliegue con el objetivo de visualizar como se desplegará el sistema desarrollado

3.2 Diagrama de despliegue

El diagrama de despliegue se utiliza para modelar la disposición física de los componentes que integran el sistema. Este diagrama describe el despliegue físico de información generada por el programa de software en los componentes de hardware [CITATION Pre10 \l 2058].

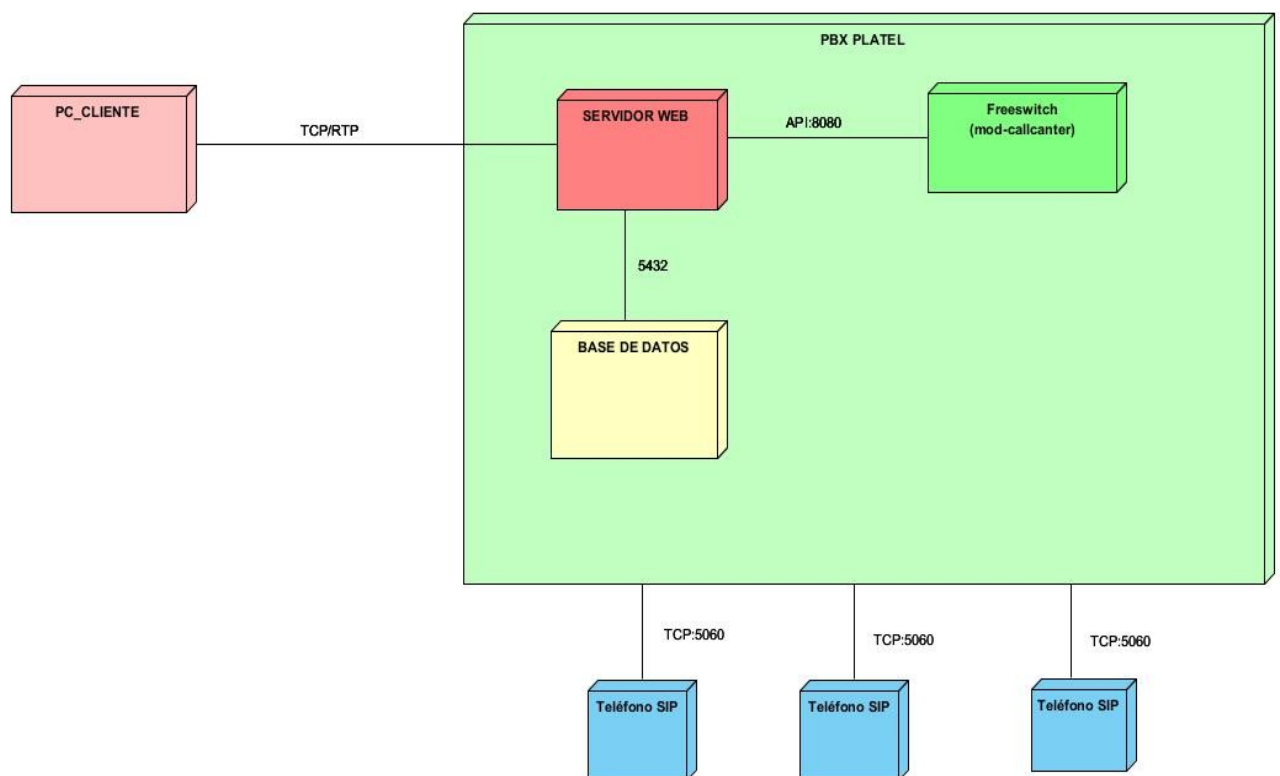


Figura 3.2. Diagrama de Despliegue

Fuente: Elaboración Propia

Descripción de elementos e interfaces de comunicación:

El proceso de solicitud al sistema se inicia con el nodo **PC_Cliente**, a través del navegador utilizando el protocolo TCP/RTP, mediante el cual se emite una petición a la PBX Platel, el cual tiene integrado un módulo de Callcenter de Freeswitch que se conecta con el servidor web a través de la API por el puerto 8080 y luego este establece una conexión con la base

de datos, la cual se encuentra integrada al servidor de aplicaciones. Los agentes se van a registrar usando un teléfono IP con el protocolo TCP:5060 a la pizarra telefónica.

Servidor Freeswitch (mod_callcenter): mod_callcenter es una aplicación de cola de llamada entrante que se puede utilizar para las necesidades del call center.

PC_Cliente: Es el nodo que se encarga de instalar el sistema con el cual el usuario interactúa desde el espacio de trabajo. Esta PC debe tener instalado el SO Windows o Linux y un navegador web, Google, Chrome o Mozilla Firefox como mínimo.

Servidor Web: Es el encargado de brindar la interfaz del sistema para que los usuarios puedan hacer uso de este. Almacena todo el código fuente del sistema y se comunica por medio del protocolo TCP con el servidor de base de datos.

Base de Datos: Almacena toda la información que brinda el sistema. La información es obtenida o modificada en dependencia del nivel de privilegio del usuario que realiza la petición. La comunicación con el servidor de aplicaciones es a través TCP empleando el puerto 5432.

Protocolo TCP: Protocolo para establecer la conexión entre el servidor de aplicaciones y el servidor de base de datos a través del puerto definido para el gestor de base de datos PostgreSQL: 5432. La conexión entre estos servidores permitirá ejecutar un conjunto de órdenes y obtener rápidamente respuesta a las mismas.

Protocolo RTP: RTP son las siglas de Real-time Transport Protocol (Protocolo de Transporte de Tiempo real). Es un protocolo de nivel de sesión utilizado para la transmisión de información en tiempo real, como por ejemplo audio y vídeo en una video-conferencia

3.3 Pruebas

En el mundo del desarrollo de software, garantizar la calidad y funcionalidad de un producto es de vital importancia. Contar con un enfoque sólido de pruebas para evitar problemas costosos y entregar software de alta calidad es, pues, fundamental para cualquier proyecto.

Las pruebas de software son un conjunto de procesos con los que se pretende probar un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento. Este tipo de pruebas abarca cualquier estado del desarrollo del sistema, desde su creación hasta su puesta en producción. Lo interesante de las pruebas es que se puedan ejecutar de manera automática, para determinar en cualquier momento si se tiene una aplicación estable o si, por el contrario, un cambio en una parte ha afectado a otras partes sin darnos cuenta[CITATION Jor22 \l 2058].

Estrategias de Pruebas

Al construir un software habitualmente se cometen errores. Es por ello que se traza una estrategia de validación que consiste en un conjunto de pruebas de software para garantizar el correcto funcionamiento y la viabilidad de la aplicación.

Una estrategia de prueba de software integra un conjunto de actividades que describen los pasos que hay que llevar a cabo en un proceso de prueba: la planificación, el diseño de casos de prueba, la ejecución y los resultados, tomando en consideración cuánto esfuerzo y recursos se van a requerir, con el fin de obtener como resultado una correcta construcción del software. En este epígrafe se muestran los resultados de la estrategia de prueba diseñada para la propuesta de solución en función de garantizar y validar la calidad de este[CITATION Pre10 \l 2058].

PRUEBAS	MÉTODO	ALCANCE
Integración	Prueba de Apis	Se realiza la integración a través de las pruebas de Apis,
Funcional	Caja negra (Partición de equivalencia)	Se validará el funcionamiento del sistema
Aceptación	Pruebas Alpha	Se aplicará en un entorno controlado con el equipo de desarrollo. El éxito de la prueba se medirá: •Exitosa: más del 75% de las funcionalidades correctas •Fracaso: menos del 75% de las funcionalidades correctas
Ejecutado por:	Glenda N. Grant Varona	

Tabla 3.1 Pruebas Realizadas

Fuente. Elaboracion Propia

3.4 Pruebas de Integración

Las pruebas de integración se pueden definir como un mecanismo de testeo de software, en el cual se realiza un análisis de los procesos relacionados con el ensamblaje o unión de los componentes, sus comportamientos con múltiples partes del sistema o de hardware. Las pruebas de integración están a cargo del examen de las interfaces entre los subsistemas o los grupos de componentes del programa o aplicación que se analiza, lo que contribuye a garantizar su funcionamiento correcto. Las “Prueba de integración” es un término amplio y abarca cualquier prueba en la que estén involucrados múltiples componentes. Las pruebas de integración son un tipo de prueba de software que garantiza que todos los componentes de las aplicaciones funcionen juntos como se espera.

En el caso de la solución desarrollada, para la verificación de una correcta interoperabilidad entre el módulo de Freeswitch y la aplicación de Monitoreo y Gestión de agentes y colas, se llevaron a cabo los casos de prueba siguientes:

Casos de Prueba	Descripción de la prueba	Resultados esperados
Adicionar un agente al sistema	Llenar los campos para adicionar un agente y hacer clic en el botón Adicionar	La aplicación crea el agente y lo añade en el mod_callcenter de Freeswitch.
Adicionar una cola al sistema	Llenar los campos para adicionar una cola y hacer clic en el botón Adicionar.	La aplicación crea la cola y lo añade en el mod_callcenter de Freeswitch.

Tabla 3.2 Casos de Pruebas

Fuente. Elaboracion Propia

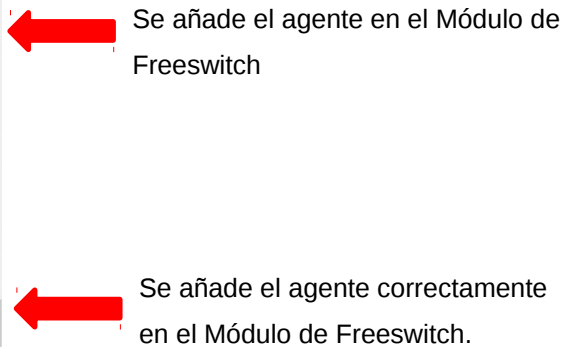
En la tabla anterior se muestra los casos de prueba utilizados para corroborar la integración del componente con la aplicación de Monitoreo y Gestión. Para desarrollar esta prueba se tuvieron en cuenta 2 escenarios que representan las posibles situaciones que se pueden presentar cuando se proceda a integrar la solución desarrollada.

A continuación, se presentarán los ejemplos de los casos de Prueba descritos anteriormente

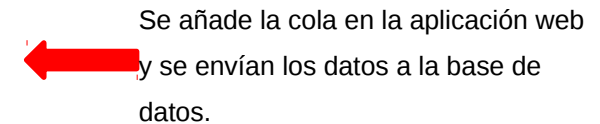
Se añade el agente en la aplicación web y se envían los datos a la base de datos.

Figura 3.3. Vista de cómo se añade el agente en la web

Fuente: Elaboración propia

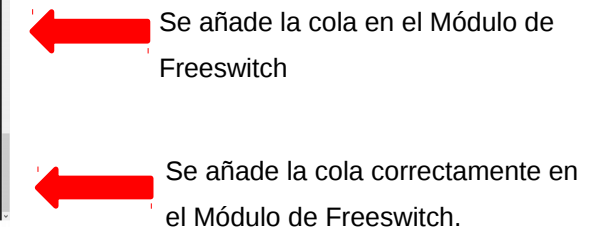


Fuente: Elaboración propia



Fuente: Elaboración propia

Fuente: Elaboración propia



Fuente: Elaboración propia

Fuente: Elaboración propia

Al integrar el módulo, se generaron las siguientes no conformidades: en la iteración 1 no existía relación entre los modelos de datos donde se encontraron 6 no conformidades asociadas a este aspecto y hacía referencia a algunas variables que no estaban definidas

en el sistema donde arrojaron 5 no conformidades, luego en la iteración 2 se arrojaron errores con las variables que estaban en español, asociadas a este aspecto se encontraron 8 no conformidades y en la tercera iteración el servidor Freeswitch API dio error porque el XML que estaba generando para agregar las colas era inactivo y arrojó 1 no conformidad.

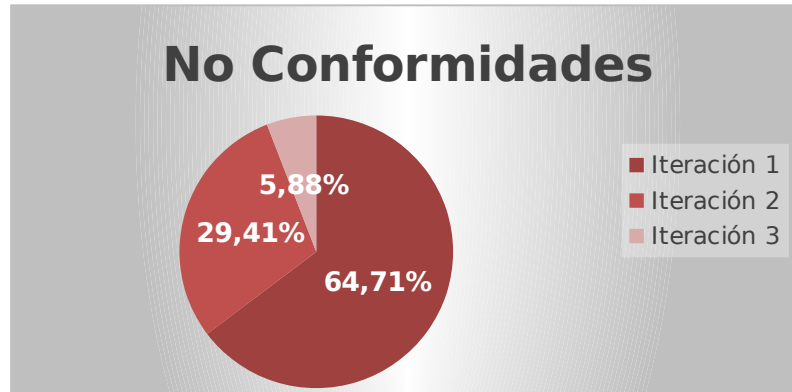


Figura 3.7. Representación gráfica de las pruebas de Integración

Fuente: Elaboración Propia

Luego de integrar el Módulo de Freeswitch con la aplicación web de Monitoreo y Gestión de agentes y colas de un centro de llamadas de la plataforma Platel, se pudo comprobar en la cuarta iteración la correcta funcionalidad del mismo con el resto de las funcionalidades que componen el sistema.

3.5 Pruebas Funcionales

Para el análisis del producto final y determinar si todas las funcionalidades son correctas se realizan las pruebas funcionales[CITATION Pre10 \l 23562]. El método utilizado para llevar a cabo este tipo de prueba es el de caja negra.

Método de Caja Negra

En la presente investigación se utilizará dentro del método de caja negra la técnica de partición de equivalencia generando los casos de prueba de dicha técnica sobre las diferentes interfaces que responden a los requisitos funcionales.

Las pruebas de caja negra se aplicaron con el objetivo de evaluar las interfaces de comunicación con el usuario, las que demostraron coherencia y funcionalidad. La técnica de partición de equivalencia es aplicada para evaluar los diferentes escenarios que pueden tener lugar ante la ejecución de una acción. Como resultado de la aplicación de estas pruebas se ejecutan las posibles variantes que posee una interfaz de comunicación con el usuario, resolviendo las no conformidades arrojadas y perfeccionando lo obtenido[CITATION Enm22 \l 23562].

Para preparar los casos de prueba es necesario un número de datos que ayuden a la ejecución de estos casos y que permitan que el sistema se ejecute en todas sus variantes. Estos datos pueden ser válidos o no para el software y son seleccionados atendiendo a las especificidades de la funcionalidad a probar. Para realizar estas pruebas existen varias técnicas:

Partición de equivalencia: divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software. Esta técnica es muy efectiva puesto que permite examinar los valores válidos e inválidos de las entradas existentes en el producto, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la realización de muchos casos antes de detectar el error genérico. La partición de equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo de este modo el número de pruebas a realizar.

Se realizaron 4 casos de pruebas a continuación se muestra un ejemplo de los mismos.

Leyenda: Estos son los campos que contiene el formulario del requisito funcional adicionar_usuario (). Más abajo aparecen en la tabla.

U: Usuario

R: Rol

C: Contraseña

H/NH: Habilitado o no Habilitado

Admin: Administrador

Escenario	Descripción	U	R	C	H/NH
EC1 Adicionar usuario con datos correctos	El administrador requiere adicionar un usuario nuevo	<u>glendangv</u> v	<u>supervisor</u> v	<u>glehb*12</u> v	<u>False</u> v
EC2 Adicionar usuario con datos incorrectos	El sistema no permite enteros sino cadena de caracteres	<u>1234678</u> F	<u>supervisor</u> v	<u>Leila*32</u> v	<u>True</u> v
EC3 Adicionar usuario con	El sistema no permite tener	<u></u>	<u>admin</u>	<u>1234aklc</u>	<u>True</u>

datos incompletos	campos vacíos	I	V	V	V
EC4 Cancelar Operación		N/A	N/A	N/A	N/A

Tabla 3.3 Casos de pruebas Funcionales

Fuente: Elaboración Propia

N°	Respuesta del sistema	Resultados de la prueba
EC1 adicionar usuario correctamente	El usuario se crea correctamente y se añade a la tabla de usuarios.	Satisfactorio
EC2 adicionar usuario con datos incorrectos	El sistema anuncia que es un campo de cadena de caracteres	No satisfactorio
EC3 adicionar usuario con datos incompletos	El campo no puede estar vacío	No satisfactorio
EC4 cancelar operación	Se muestra la interfaz de usuarios del sistema.	No satisfactorio

Tabla 3.4 Casos de pruebas Funcionales

Fuente: Elaboración Propia

Resultado de las pruebas funcionales

Después del análisis de los escenarios de pruebas se detectaron en las 3 iteraciones realizadas un total de 16 no conformidades. En la primera iteración se obtuvieron 12 no conformidades que fueron resueltas, la segunda iteración arrojó 4 no conformidades que fueron solucionadas y para una tercera iteración y final no se detectaron no conformidades. En la Figura 13 se muestran los resultados por cada iteración realizada al sistema.

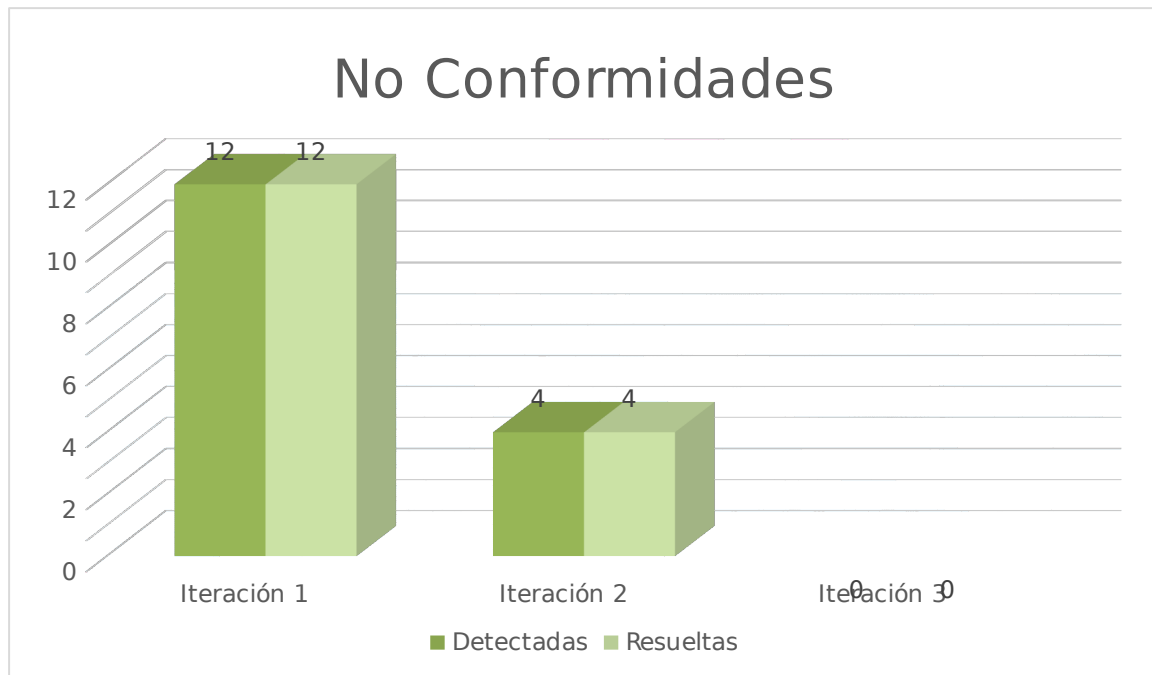


Figura 3.8. No Conformidades de las pruebas Funcionales

Fuente: Elaboración Propia

Las no conformidades identificadas durante el proceso de pruebas son:

Los datos incorrectos introducidos por los usuarios son guardados en las bases de datos sin realizar una validación previa donde se encontraron 4 no conformidades referentes a este aspecto. Esta no conformidad se obtenía por la validación de campos en blanco y fue solucionada al añadir una aserción para que estos campos no validaran y guardaran campos vacíos.

Presencia de errores ortográficos donde se encontraron 12 no conformidades referente a este aspecto. Esta no conformidad fue resuelta al corregir los problemas de acentuación en las etiquetas de los campos y en las notificaciones de usuario.

3.6 Pruebas de Aceptación

Las pruebas de aceptación, son supervisadas por el cliente basándose en los requerimientos tomados de las historias de usuario. En todas las iteraciones, cada una de las historias de usuarios seleccionadas por el cliente deberá tener una o más pruebas de aceptación, de las cuales deberán determinar los casos de prueba e identificar los errores que serán corregidos. Las pruebas de aceptación son pruebas de caja negra, que representan un resultado esperado de determinada transacción con el sistema[CITATION Rog10 \l 2058].

Para la revisión final de las especificaciones del diseño y de la implementación se realizaron las pruebas de aceptación con el objetivo de determinar diferentes tipos de errores y resolverlos para ello se emplean las pruebas Alfa y Beta.

- ✓ **Pruebas Alfa:** Se desarrollan en conjunto, el desarrollador y los usuarios finales. Con el objetivo de registrar los errores y problemas del uso del software.
- ✓ **Pruebas Beta:** Se realiza en el sitio final donde se va a desplegar el software. A diferencia de la prueba alfa, la prueba beta es una aplicación del software en su ambiente final. El objetivo es registrar todos los problemas que se encuentran durante la prueba.

El objetivo de estas pruebas es verificar que el software está listo y que puede ser usado por los usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

Como resultado después de haber aplicado las pruebas Alfa se identificaron nuevas no conformidades, las cuales fueron resueltas. Las pruebas Alfas fueron aplicadas en un ambiente controlado, donde se probó por parte del equipo de desarrollo la funcionalidad del sistema, arrojaron errores en el cumplimiento de los requisitos funcionales requeridos para el sistema. El cliente emitió un acta de aceptación de los productos de trabajo en total conformidad con la solución desarrollada (ver Anexo 3).

3.7 Conclusiones parciales

En el presente capítulo se:

1. Se realizó la confección de un modelo de despliegue el cual permitió describir la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo.
2. Se expuso el diagrama de componentes del sistema lo que propició una mejor comprensión de la estructura arquitectónica de la solución desarrollada.
3. La aplicación de pruebas de software permitió identificar y corregir errores cometidos en la implementación del sistema y se obtuvo una aplicación que cumple con los requisitos funcionales determinados por el cliente.

Conclusiones generales

Para dar cumplimiento a los objetivos propuestos en la presente investigación, se concluye lo siguiente:

1. El análisis y elección tanto de las herramientas, tecnologías, así como la metodología escogida para desarrollar la presente investigación, permitieron abordar un poco más sobre las posibles propuestas de solución para dar respuesta al problema planteado, así también se analizaron las soluciones similares existentes y se determinaron que no cumplen completamente con los requerimientos de la aplicación propuesta.
2. La relación existente entre varias áreas de la programación y la ingeniería de software, permitió realizar de manera exitosa el correcto diseño e implementación de la aplicación de Monitoreo y Gestión de agentes y colas del centro de llamadas de la plataforma Platel.
3. Con el desarrollo de la aplicación web de Monitoreo y Gestión de agentes y colas del centro de llamadas de la plataforma Platel se logró supervisar de forma concurrente todos los eventos asociados al monitoreo de agentes y colas, dejando de existir el retardo en la visualización de los eventos ocurridos y de esta forma se muestran los resultados actualizados.
4. La estrategia de pruebas definida, junto a la validación de las mismas permitieron obtener resultados positivos acerca de la propuesta de solución desarrollada, logrando que la aplicación cumpla con los requisitos definidos planteados por el cliente.

Recomendaciones

- ✓ Incluir el análisis de datos en el centro de llamadas para incorporar centro de mandos y una mejor toma de decisiones.
- ✓ Permitirle al supervisor contestar una llamada de una cola determinada si no hay agentes disponibles y escuchar la conversación de un agente que está atendiendo una llamada en una cola.

Referencias Bibliográficas

3CX. 2023. [En línea] 2023. <https://www.3cx.es/voip-sip/voip-definicion/>.

Alayo, Yanisleydi Campos. 2013. *Modulo Distribuidor Automatico de Llamadas para YATE*. 2013.

Anthony. 2018. *FreeSWITCH 1.10*. 2018.

Arianna. 2015. *Aplicación para monitorear en tiempo real las colas, extensiones y*. 2015.

Barajas, Rodriguez. 2018. *Impacto de los requerimientos en la calidad de software | Tecnología Investigación y Academia*. 2018.

bootstrap. 2022. *Bootstrap: qué es y cómo funciona este framework*. 2022.

Botero, Toro Lazo y Gálvez. 2016. *Especificacion de requisitos de software: una mirada desde la revisión teórica de antecedentes. Entre Ciencia e Ingeniería*, vol. 10, no- 19, pp. 108-115. ISSN 2539-4169. 2016.

Bridge, Walter. 2021. *Glosario definitivo para profesionales de Call Center, Contact Center y Experiencia de Cliente*. 2021.

BRIDGE, WALTER. 2021. *Glosario definitivo para profesionales de Call Center, Contact Center y Experiencia de Cliente*. 2021.

Buchillon, Fernandez. 2019. *Sistema para le Gestion de la Prenomina del Centro de Ideoinformatica*. 2019.

Buscatto. 2002. *Les centres d'appels, usines modernes? Les rationalisations*. 2002.

2023. Call Center. [En línea] 2023. <https://definicion.de/call-center/>.

Canelo, Miriam Martínez. 2020. *Patrones de diseno de software*. 2020.

Carbó, Yosvany Medina. 2016. *Cuba y el impacto de las TIC en la informatización de la sociedad*. 2016.

Castillo. 2009. *Distribuidor Automático de Llamadas (ACD) para PLATEL*. 2009.

Chile., IDA. 2016. *Modelos conceptuales para definir las plataformas digitales*. 2016.

Cristina, Sara. 2016. *PROPUESTA DE MEJORA EN LOS PROCESOS DEL ÁREA DE CALL CENTER TÉCNICO DE UNA EMPRESA DE TELECOMUNICACIONES*. 2016.

cubadebate. 2020. *Etecsa informa sobre nueva oportunidad para desarrolladores a través de Apklis.* 2020.

—. **2022.** *Universidad de las Ciencias Informáticas: 20 años del sueño de Fidel (+ Fotos y Video).* 2022.

ecured. 2023. *Empresa de Tecnologías de la Información para la Defensa.* 2023.

—. **2023.** *Estrategia de pruebas de software.* 2023.

Ferrer, Jaime Moya. 2019. *Análisis de Herramientas de Gestión de VOIP.* 2019.

Grech, M. 2018. *The Comprehensive Guide To Understanding VoIP Protocols and Standards.* 2018.

Hernández, Arianna de Arma. 2019. *Aplicación web para la gestión de la información especializada en Geociencia.* 2019.

Higo. 2021. *Protocolo de comunicación: ¿Qué es y para qué sirve?* 2021.

IEEE. 2002. [En línea] JUNIO de 2002. <https://www.ietf.org/rfc/rfc3261>.

Ignacio. 2012. *La experiencia del cliente desde el punto de vista del contact center.* 2012.

Internet Engineering Task Force (IETF). 2021. [En línea] enero de 2021. <https://datatracker.ietf.org/doc/html/rfc8835>.

JavaScript. 2023. <https://developer.mozilla.org/es/docs/Web/JavaScript>. 2023.

Juan. 2019. *OpenWebinars.* 2019.

Julián Yanover. 2023. *Call Center.* [En línea] 2023. <https://definicion.de/call-center/>.

LÓPEZ, EDDY SANTIAGO GONZÁLEZ. 2018. *SOLUCIÓN DE VOIP CON SERVIDORES FREEPBX A NIVEL EMPRESARIAL EN BOGOTÁ .* 2018.

Luis. 2015. *Módulo para la gestión de respuesta de voz interactiva (IVR) en la plataforma telefónica Platel.* 2015.

M. Rouse. 2008. *IAX (Inter-Asterisk Exchange Protocol).* 2008.

Mark.J. 2019. *Bienvenido a FusionPBX Docs.* 2019.

Matango, Franklin. 2016. *Tecnologías comerciales existentes de VoIP.* 2016.

McPhail. 2002. *“What is ‘on the line’ in call centre studies?: A review of key.* 2002.

- Míguez, Juan. 2020.** ¿Qué es un códec de VoIP y por qué es importante? [En línea] 8 de septiembre de 2020. <https://centralitasbaratas.es/que-es-un-codec-de-voip-y-por-que-es-importante/>.
- Montero, Sergio Infante. 2013.** Curso Django: Entendiendo como trabaja Django. [En línea] 2013. <https://www.maestrosdelweb.com/curso-django-entendiendo-como-trabaja-django/>.
- Mordor. 2020.** *Unified Communications as a Service*. 2020.
- OPENAI. 2023.** *Descripción de los componentes de un diagrama de componentes basado en el patrón arquitectónico MTV*. 2023.
- Ortega, Gilberto Andres Vargas. 2021.** *Lineamientos para el diseño de aplicaciones web soportados en patrones GRASP*. Ciencia e Ingeniería. 2021, Vol. Vol 8. . 2021.
- Pandya, R. 1995.** *Emerging mobile and personal communication systems*. 1995.
- Pressman. 2010.** *Ingeniería del Software. Un enfoque práctico*. Quinta Edición. Mc Graw Hill. 2010.
- Pressman, Roger. 2010.** *Ingeniería del software, Un enfoque practico*. Séptima edición. Pressman. 2010.
- Quarea. 2019.** *QueueMetrics: Estadísticas de recepción llamadas ACD*. 2019.
- QueueMetrics. 2023.** *Centro de Llamadas Asterisk Software de Monitoreo*. 2023.
- Rodríguez, Angela Maria Bauta. 2021.** *Diseño de Sistema para Gestionar la Información de los Expedientes de Cuadros en la Oficina Nacional de Administración Tributaria*. 2021.
- Rubén, Jandry. 2021.** *El avance tecnológico en la telefonía móvil* . 2021.
- Rubio, Javier Cerezo. 2023.** *Diseño de una Aplicación para Consulta de Datos de Pacientes Con Cáncer de mama*. 2023.
- Sánchez, Tamara Rodríguez. 2015.** *Metodología de desarrollo para la Actividad pro-*. 2015.
- Sánchez Maldonado, Winston Alexander. 2018.** *Análisis de opciones de acceso desde Internet al servicio de comunicaciones unificadas con Cisco*. 2018.
- Silva, Yaneysi Velázquez. 2022.** *Ciencias y Tecnologías Computacionales*. 2022.
- SinoLogic. 2010.** *Todo lo que has querido saber de DAHDI*. 2010.

Sommerville. 2006. *Ingenieria de Software - Ian Sommerville 7a Edicion.* 2006.

TANYA, MICHELLE y. 2018. *PROPUESTA DE OPTIMIZACIÓN DEL SERVICIO AL CLIENTE EN LAS.* 2018.

Turrado, Jorge. 2020. *Qué son las pruebas de software.* 2020.

Vera, Dariel López. 2022. *Sistema para la gestión de contenido de la.* 2022.

Vilchez, Ruiz. 2023. *La innovación tecnológica y el cambio regulatorio en telecomunicaciones.* 2023.

Vitón, Enmanuel Díaz. 2022. *Sistema Infomatico para la Gestion de la Informacion asociada a las Catedras Honorificas en la UCI.* 2022.

VozToVoice. 2020. *Interfaz grafica para Freeswitch.* 2020.

Webopedia, E. 2018. *The Difference Between VoIP and PSTN Systems.* 2018.

XETID. 2023. *XETID, empresa cubana líder en el desarrollo de.* 2023.

Xetid. 2023. *XETID, empresa cubana líder en el desarrollo de soluciones tecnicas para la informatizacion de la sociedad.* 2023.

Yasmin. 2022. *Aplicación web para la gestión de.* 2022.

ANEXOS

Anexo 1

Entrevista realizada al equipo de análisis del proyecto PLATEL

A continuación, se presenta el cuestionario para esta entrevista:

1. ¿Cuál es su percepción sobre la importancia del uso de la aplicación web que monitorea y gestiona los agente y las colas en la plataforma PLATEL y por qué cree que es relevante?
2. ¿Cuáles son los objetivos que se persiguen al utilizar esta aplicación web en el centro de telecomunicaciones de XETID?
3. ¿Podría describir el proceso actual de configuración del centro de llamadas de la Plataforma PLATEL?
4. Basándose en su experiencia, ¿Considera necesario contar con una aplicación web de monitoreo y gestión de agentes y colas? ¿Qué beneficios aportaría?
5. ¿Existen restricciones o regulaciones específicas que la aplicación web debe cumplir, por ejemplo, en relación con la privacidad y protección de datos personales?

Anexo 2

<i>Historia de Usuario # 16:</i>	
Número: 16	Nombre del requisito: Modificar Cola
Programador: Glenda Grant Varona	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en Desarrollo: Alta	Tiempo Real: 5 días
Descripción: Esta funcionalidad debe permitir al administrador modificar las colas del sistema.	
Observaciones: Se deben manejar todas las restricciones requeridas por el cliente.	

<i>Historia de Usuario # 17:</i>	
Número: 17	Nombre del requisito: Eliminar Cola

Programador: Glenda Grant Varona	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en Desarrollo: Alta	Tiempo Real: 5 días
Descripción: Esta funcionalidad debe permitir al administrador eliminar las colas del sistema.	
Observaciones: Se deben manejar todas las restricciones requeridas por el cliente.	

Historia de Usuario # 14:	
Número: 14	Nombre del requisito: Listar Cola
Programador: Glenda Grant Varona	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en Desarrollo: Alta	Tiempo Real: 5 días
Descripción: Esta funcionalidad debe permitir al administrador listar las colas del sistema.	
Observaciones: Se deben manejar todas las restricciones requeridas por el cliente.	

Historia de Usuario # 11:	
Número: 11	Nombre del requisito: Modificar Agente
Programador: Glenda Grant Varona	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 4 días
Riesgo en Desarrollo: Alta	Tiempo Real: 4 días
Descripción: Esta funcionalidad debe permitir al administrador modificar los agentes del sistema.	
Observaciones: Se deben manejar todas las restricciones requeridas por el cliente.	

Historia de Usuario # 12:	
Número: 12	Nombre del requisito: Eliminar Agente
Programador: Glenda Grant Varona	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 4 días
Riesgo en Desarrollo: Alta	Tiempo Real: 4 días
Descripción: Esta funcionalidad debe permitir al administrador eliminar los agentes del sistema.	
Observaciones: Se deben manejar todas las restricciones requeridas por el cliente.	

Historia de Usuario # 9:	
Número: 9	Nombre del requisito: Listar Agente
Programador: Glenda Grant Varona	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 4 días
Riesgo en Desarrollo: Alta	Tiempo Real: 4 días
Descripción: Esta funcionalidad debe permitir al administrador listar los agentes del sistema.	
Observaciones: Se deben manejar todas las restricciones requeridas por el cliente.	

Historia de Usuario # 22:	
Número: 22	Nombre del requisito: Adicionar Centro de Llamadas
Programador: Glenda Grant Varona	Iteración Asignada: 3
Prioridad: Alta	Tiempo Estimado: 6 días
Riesgo en Desarrollo: Alta	Tiempo Real: 6 días
Descripción: Esta funcionalidad debe permitir al administrador añadir los centros de llamadas en el sistema.	

Observaciones:

Se deben manejar todas las restricciones requeridas por el cliente.

Historia de Usuario # 20:

Número: 20	Nombre del requisito: Modificar Centro de Llamadas
Programador: Glenda Grant Varona	Iteración Asignada: 3
Prioridad: Alta	Tiempo Estimado: 6 días
Riesgo en Desarrollo: Alta	Tiempo Real: 6 días
Descripción: <i>Esta funcionalidad debe permitir al administrador modificar los centros de llamadas en el sistema.</i>	
Observaciones: <i>Se deben manejar todas las restricciones requeridas por el cliente.</i>	

Historia de Usuario # 19:

Número: 19	Nombre del requisito: Listar Centro de Llamadas
Programador: Glenda Grant Varona	Iteración Asignada: 3
Prioridad: Alta	Tiempo Estimado: 6 días
Riesgo en Desarrollo: Alta	Tiempo Real: 6 días
Descripción: <i>Esta funcionalidad debe permitir al administrador listar los centros de llamadas en el sistema.</i>	
Observaciones: <i>Se deben manejar todas las restricciones requeridas por el cliente.</i>	

Anexo 3



FACULTAD DE CIENCIAS Y TECNOLOGÍAS COMPUTACIONALES
VICEDECANATO DE FORMACIÓN

AVAL DEL CLIENTE

Título: Monitoreo y Gestión de agentes y colas del centro de llamadas de la plataforma Platel

Autor(es): Glenda Nubia Grant Varona

La Parte Cliente, luego de haber revisado artefactos generados y la aplicación web determina que Glenda Nubia Grant Varona ha finalizado su aplicación web Monitoreo y Gestión de agentes y colas del centro de llamadas de la plataforma Platel.

Este proyecto fue apoyado metodológicamente y técnicamente en la fase de planificación, ejecución, procesamiento, análisis e interpretación de los datos, así como sus respectivas conclusiones. Después de revisar la resolución de los requisitos y la incorporación de las observaciones brindadas por la parte cliente y consultas realizadas a especialistas en el tema, se considera que el mismo hace una aportación significativa al conocimiento y le da un valor agregado a la Plataforma Platel al incorporarle un nuevo módulo que va a contribuir a la informatización de la sociedad cubana. Además, las empresas que la usen garanticen la satisfacción del cliente brindando un servicio de soporte técnico, atención al cliente y resolución de problema de forma rápida y satisfactoria.

Por consiguiente, se valora que la misma cumple con todos los requisitos y objetivos propuesto y, por tanto, está listo para su explotación experimental.

Nombre y Apellidos: Ronney Zulueta Esquivel

Cargo: Especialista Superior en TIC

Firma:

Fecha: 08/11/2023