Universidad de las Ciencias Informáticas

Empresa de Tecnologías de la Información para la Defensa

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

---

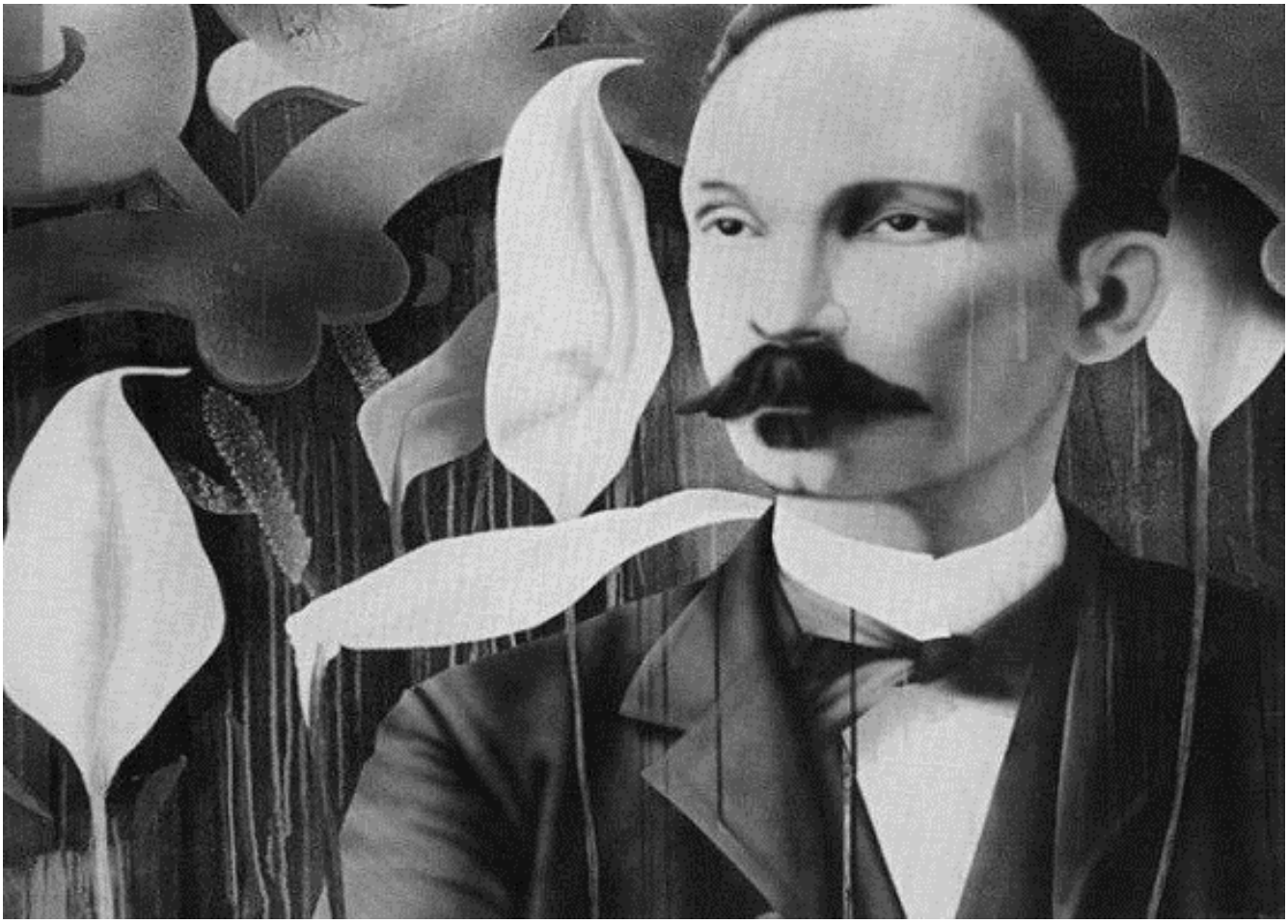
Título: Componente de comunicación sincrónica para la Plataforma de Gobierno Electrónico “Bienestar”.

Autor: Rolando Ramírez Hidalgo

Tutores: Ing.Raciel Garrido Torres

Ing. Niurka Socarrás Hernández

La Habana,18 de junio de 2022



“Haga cada uno su parte de deber y nada podrá vencernos”

JoséMartí

Declaración de autoría

Declaro por este medioque yo,Rolando Ramírez Hidalgo, con carné de identidad 95092616747, soy el autor principal del Trabajo Final de Tesis de Pregrado que se titula: Componente de comunicación sincrónica para la Plataforma de Gobierno Electrónico “Bienestar”. El cual ha sido desarrollado como parte del trabajo en la divisiónde Gobierno Electrónico de la Empresa de Tecnologías de la Información para la Defensa(XETID). Autorizo a la XETIDa hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, se firma la presente declaración jurada de autoría en La Habana el día 18 de junio de 2022.

_____

Rolando Ramírez Hidalgo

Firma del Autor

_____ _____

Ing. Raciel Garrido TorresIng. Niurka Socarrás Hernández

Firma del TutorFirma del Tutor

Agradecimientos

A mis padres por haberme apoyado durante estelargo trayecto en busca de lograr mi sueño. Por darme su amor y cariño en los buenos momentos y en lasmás difíciles circunstancias, por su educación, por sus valores y sus consejos.

A mis tutores, por su consagración y apoyo para que este trabajo resultara lo mejor posible.

A mis compañeros de aula, con los cuales he recorrido estos cinco años.

Al movimiento ICPC, donde me hice de grandes amigos y sólidos conocimientos.

Dedicatoria

A mis padres (Rolando y Ania) por ser mi ejemplo a seguir y apoyarme entodo, de verdad me siento privilegiado de ser su hijo.

A mis hermanitas (Kathy y Patry) por ser mi razón de ser.

Resumen

La Empresa de Tecnologías de la Información para la Defensa (XETID) trabaja en el proceso de informatización de las instituciones de gobierno. Para ello ha desarrollado la Plataforma de Gobierno Electrónico “Bienestar”, que ofrece transparencia en la información asociada a los diferentes procesos de los servicios públicos.

La Plataforma “Bienestar” no ofrece una funcionalidad que permita a sus usuarios comunicarse de forma sincrónica. Para realizar dicha acción hay que acudir a otras soluciones como los clientes demensajería instantánea ajenos al sistema o al uso de latelefonía convencional. El uso de estas aplicaciones y otros medios de comunicaciónalternativos traen consigo gastos adicionales y pérdidas de tiempoque limitan y disminuyen las potencialidades de la Plataforma,afectando así los procesos que en ella se desarrollan,demorando la toma de decisiones y creando descontento en la población.

El presente trabajo tiene como objetivo desarrollar un componente que al ser integrado a la Plataforma permitael intercambio de mensajes de texto en tiempo real y la realización de videollamadas entre sus usuarios. Para su desarrollo se emplean tecnologías VOIP aprovechando los servicios ofrecidos por la Plataforma de Telecomunicaciones “Platel” desarrollada por la XETID, seemplea la metodología PRODESOFT para guiar el proceso de desarrolloy UML comolenguaje de modelado, PhpStorm como entorno dedesarrollo integrado y Gulliver como marco de trabajo. También se utilizan PHP y JavaScript comolenguajes de programación. Finalmente fueronimplementadas las funcionalidades definidas y realizadas sus pruebas correspondientes, obteniéndoseresultados satisfactorios.

Palabras claves:gobierno electrónico, comunicación, VoIP, video, voz, mensajería.

Abstract

The Defense Information Technology Company (XETID) works in the computerization process of government institutions. To this end, it has developed the “Bienestar” Electronic Government Platform, which offers transparency in the information associated with the different processes of public services.

The “Bienestar” Platform does not offer functionality that allows its users to communicate synchronously. To carry out this action, you have to resort to other solutions such as instant messaging clients outside the system or the use of conventional telephony. The use of these applications and other alternative means of communication bring with them additional expenses and time losses that limit and diminish the potential of the Platform, thus affecting the processes that take place in it, delaying decision-making and creating discontent among the population.

The objective of this work is to develop a component that, when integrated into the Platform, allows the exchange of text messages in real time and video calls among its users. For its development, VOIP technologies are used taking advantage of the services offered by the "Platel" telecommunications platform developed by XETID, the PRODESOFT methodology is used to guide the development process and UML as a modeling language, PhpStorm as an integrated development environment and Gulliver as a framework. PHP and JavaScript are also used as programming languages. Finally, the defined functionalities were implemented and their corresponding tests were carried out, obtaining satisfactory results.

Keywords: electronic government, communication, VoIP, video, voice, messaging.

Índice de contenidos

Introducción

El avance de las Tecnologías de la Información y las Comunicaciones (TIC) en el mundo y en Cuba ha incentivado a la dirección del país a encaminar la implementación de un Gobierno Electrónico queestimuleel uso de las TIC en los órganos de la Administración para mejorar la información y los servicios ofrecidos a los ciudadanos, orientar la eficacia y eficiencia de la gestión pública e incrementar sustantivamente la transparencia del sector público y la participación de los ciudadanos[1].El proceso de informatización de la sociedad pasa por evolucionar desde una perspectiva centrada en los procesos y sistemas internos de las entidades gubernamentales, hacia un nuevo modelo centrado en los ciudadanos.

Las instituciones de gobierno con su rol de servidores públicos, lideran procesos de transformación debidos a los cambios sociales, económicos, políticos y tecnológicos. Estos procesos determinan las líneas de actuación y los retos a los que se enfrentan los diferentes niveles administrativos en los que se organiza el estado, priorizando que permitan un ahorro en tiempo, esfuerzo y costos, de forma que posibilite mejorarsu trabajo y brindar la posibilidad a los ciudadanosa relacionarse con las diferentes administraciones y obtener un servicio en tiempo y forma, ajustado a su necesidad y con un alto nivel de calidad.

La Empresa de Tecnologías de la Información para la Defensa (XETID) trabaja en el proceso de informatización de las instituciones de gobierno para la implantación de distintas soluciones que les ayuden en su mejora continua. Para ello ha desarrollado la Plataforma de Gobierno Electrónico “Bienestar” (en lo adelante Plataforma “Bienestar”), que ofrece transparencia en la información asociada a los diferentes procesos de los servicios públicos ybrinda una mejor experiencia al ciudadano en su participación y gestión con las entidades del gobierno.

La Plataforma “Bienestar”permite a través de un catálogo de servicios digitales que las entidades del gobierno puedan hacer uso de las TIC; para llevar a cabo una administración más eficiente, posibilitando centralizar todos los procesos, trámites y gestiones empresariales en una sola instancia, con el objetivo de racionalizar y simplificar dichos procesos al tiempo que favorece la integración entre las organizaciones que tienen relaciones afines.

Actualmente las vías de comunicación que brinda la Plataforma “Bienestar” entre las entidades que la utilizan son mediante notificaciones por el correo electrónico y las actividades que deben realizar en la misma plataforma, pero no se cuenta con una solución de comunicación sincrónica que facilite o mejore la interacción entre ellas. Por lo que la falta de comunicación sincrónica entre los funcionarios del gobierno, las entidades administrativas e incluso los ciudadanos dentro de la Plataforma

“Bienestar”dificulta que se creen estructuras de cooperación efectivas entre las distintas instituciones y los ciudadanos, obligando a buscar terceras vías de comunicación como recurrir a la telefonía convencional acarreando en costos adicionales en pagos de facturas telefónicas y pérdida de tiempo por lo que la eficiencia y calidad de servicio puede verse afectada, demorando la toma de decisiones y creando descontento en la población.

Ante esta situación se plantea como **problema de investigación**: ¿cómo facilitar la comunicación sincrónicaentre los funcionarios del gobierno, las entidades administrativas y los ciudadanos dentro de la Plataforma “Bienestar”?

Para dar solución a este problema se tomará como **objeto de estudio**: los sistemas de comunicación sincrónica.Tomando como **campo de acción**: los sistemas de comunicaciónsincrónica embebidos entecnologías webs.

Teniendo en cuenta lo antes planteado, el **objetivo general** de esta investigación es: desarrollar un componente para la comunicación sincrónicaen la Plataforma “Bienestar” que permita mayor interacción entre los funcionarios del gobierno, las entidades administrativas y los ciudadanos.

Para cumplir con el objetivo general expuesto, se han trazado varios **objetivos específicos:**

1) Fundamentar los antecedentes y aspectos teóricos referentes a los sistemas de comunicación sincrónica.
2) Modelar un componente de comunicación sincrónica para la Plataforma “Bienestar”.
3) Implementar un componente de comunicación sincrónica para la Plataforma “Bienestar”.
4) Validar el correcto funcionamiento del componente realizado mediante pruebas de software y juicio de expertos.

Con el propósito de cumplir con los objetivos planteados se establecen las siguientes **tareas de la investigación**:

1) Elaborar el marco teórico conceptual de la investigación.
2) Definición de los métodos y técnicas de investigación a emplear.
3) Caracterización del estado del arte de sistemas similares.
4) Descripción del entorno y la metodología de desarrollo a emplear.
5) Definición de los requisitos funcionales y no funcionales según el proceso de desarrollo aplicado.
6) Validación de los requisitos obtenidos aplicando técnicas afines.
7) Desarrollo del componente.
8) Elaboración de pruebas aplicables al componente.

9) Análisis de los resultados de las pruebas realizadas.

Los **métodos de investigación** utilizados para llevar a cabo la presente investigación son:

Métodos teóricos:

- **Modelación:**facilitó la representación, mediante el uso de diagramas, dela realidad de la investigación, los principales elementos que la componen y su funcionamiento, lo cual ayudó a descubrir y estudiar nuevas cualidades y relaciones del objeto de estudio.
- **Analítico-Sintético:** mediante la lectura y comprensión de la documentación estudiada permitióadquirir losconocimientos necesarios para el desarrollo del componente.

Métodos empíricos:

- **Entrevista:** se concretaron encuentros informales con el cliente mediante los cuales sepudo definir la necesidad de la creación del componente, establecer requisitos funcionales, observar elavance durante las fases de desarrollo y validación.

El presente trabajo se estructura en tres capítulos, los cuales son:

Capítulo 1: Fundamentación teórica

Se definen algunos conceptos importantes para la posterior comprensión de la investigación. Se realizauna breve reseña y valoración sobre las soluciones existentes de los sistemas de comunicación sincrónica, también se explican las metodologías,tecnologías y herramientas que se utilizan en el desarrollo de la solución.

Capítulo 2: **Modelación de la solución**

Se brinda una explicación de la solución propuesta, enfatizando las características del componente adesarrollar. Se detallan los artefactos generados de acuerdo a la metodología PRODESOFT durante su fase de modelación.

Capítulo 3: Construcción de la solución

Se definen los estándares de codificación que debe cumplir el equipo de desarrollo. Se detallan los artefactos generados durante la fase de construcción de la metodología PRODESOFT. Se valida la solución obtenida mediante la realización de pruebas de software.

Capítulo 1: Fundamentaciónteórica

1.1 Introducción

En el presente capítulo se expone la fundamentación teórica del trabajo de diploma, incluyéndose en el mismo el estudiodeotras soluciones informáticas existentes en la actualidad que resuelven la problemática planteada.También se exponen las metodologías, tecnologías y herramientas de desarrollodesoftware a utilizar en el desarrollo de la solución.

1.2 Conceptos fundamentales

A continuación, se presentan un grupo de conceptos que se hacen necesarios para la comprensión de la presente investigación.

1.2.1 Comunicación

El concepto de comunicación ha sido formulado por varios autores, dentro de las contribuciones encontradas resalta la siguiente:

La comunicación es el proceso mediante el cual se puede transmitir información de una entidad a otra independientemente del espacio temporal donde se encuentren. El acto de comunicar radica en la interacción entre dos o más personas que realizan el intercambio de mensajes, utilizando un canal que sirve de soporte en la transmisión de la información. Todas las formas de comunicación requieren un emisor, un mensaje y un receptor[2].

A partir de este criterio se puede conformar el siguiente concepto de comunicación sincrónica que regirá la investigación:

La **comunicación sincrónica** es el intercambio de informaciónque se realiza entre dos o más personas en un mismo espacio temporal donde la información intercambiada está compuesta por texto, audio y/o video y el canal que sirve de soporte en la transmisión es Internet mediado por computadoras.

1.2.2 Voz sobre protocolo de internet(VoIP)

Es un conjunto de tecnologías que hacen posible que la señal de voz viaje a través de Internet empleando el protocolo IP (Protocolo de Internet). Esto significa que se envía la señal de voz en forma digital, en paquetes de datos, en lugar de enviarla en forma analógica a través de circuitos utilizables solo por telefonía convencional[3].

A partir del concepto de comunicación sincrónica se plantea que VoIP es el conjunto de tecnologías que permiten la misma a través del protocolo IP.

1.3 Plataforma de Gobierno Electrónico "Bienestar"

La Plataforma "Bienestar"constituye una solución integral basada principalmente en funcionalidades ofrecidas por tecnologías de software libre. Las mismas permiten la informatización de los procesos que benefician la gestión de los trámites de los ciudadanos con las entidades gubernamentales y administrativas.A continuación, se describen los sistemas y componentes que forman parte de la solución:

- **Portal de Servicios Integrales:** sitio web para acceder a los trámites disponibles en la entidad.Desarrollado sobre tecnologías soportadas por *WordPress* con el objetivo de brindar una interfaz de usuario agradable, gestionable e intuitiva, garantizando su usabilidad en dispositivos móviles u otro medio portátil.
- **Catálogo de Trámites:** componente que funciona como repositorio de trámites y/o procesos previamente modelados. La garantía de contar con un catálogo de trámites brinda la posibilidad de reutilizar el conocimiento adquirido en la entidad para dar respuestas a las necesidades de los clientes.
- **Motor de Tramitación:** posibilita la gestión y configuración de los procesos y trámites en la Plataforma. Basado en las funcionalidades ofrecidas en "*ProcessMaker*" y WSO2[1], el *Motor de Tramitación* brindará la posibilidad de transformar la entidad en una organización cuya gestión se realiza por integración de procesos y servicios, lo que constituye un cambio de paradigma.
- **Gestor Documental (Dfile):** posibilita la gestión del proceso documental asociados a los servicios y trámites que ofrece la entidad en la ejecución de sus procesos. El gestor documental garantiza seguridad, trazabilidad, auditoría, monitoreo y control de la documentación (física y documental) de la entidad. Basado en el software libre Alfresco.
- **Diseñador de Servicios SOA[2]:** posibilita gestionar la capa de servicio de la plataforma para garantizar interoperabilidad entre los sistemas que conforman la solución integral. El diseñador de servicios posibilitará exportar la lógica de negocio a una capa de servicios que cumpla con estándares SOA para la integración de sistemas haciendo uso de tecnologías WSO2.

[1] Es una compañía que desarrolla aplicaciones de software abierto enfocadas en proveer una arquitectura orientada a servicios (SOA) para desarrolladores profesionales.
[2] Arquitectura Orientada a Servicios (SOA, siglas del inglés Service Oriented Architecture).

A continuación, se muestra una vista en tres capas de la plataforma donde se pueden apreciar los tres sistemasde código abierto en que se basa la solución:

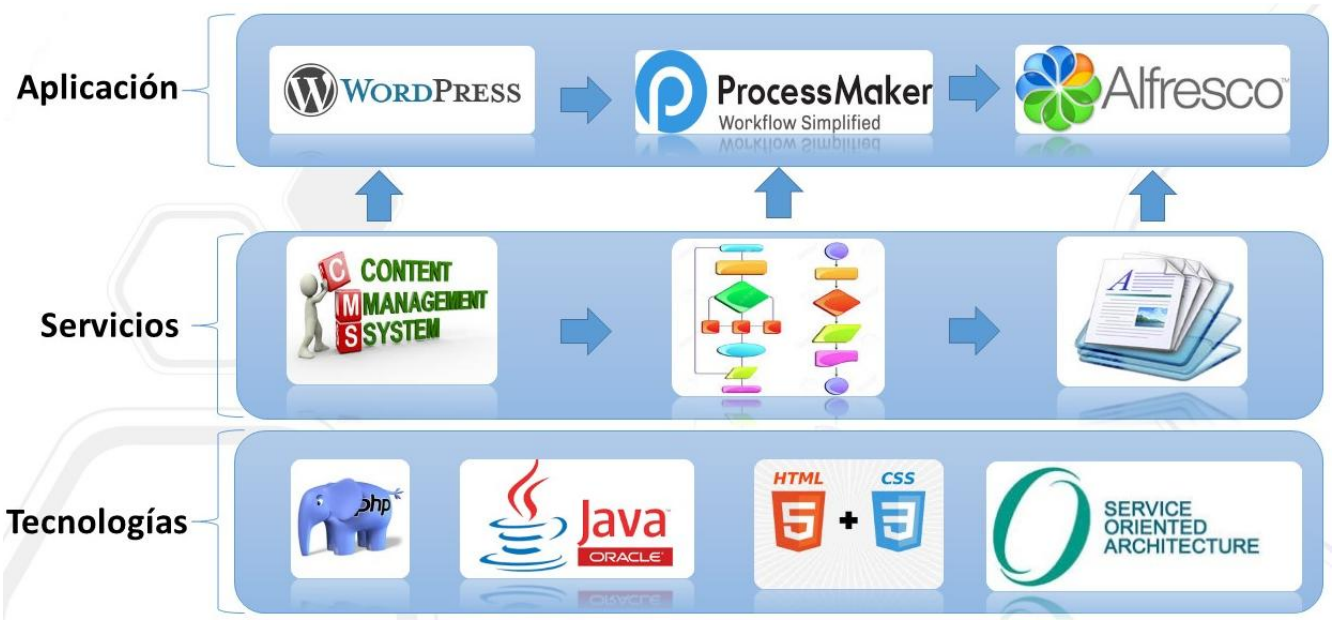


Figura 1: Vista en tres capas de la Plataforma ""Bienestar""

- **WordPress:**esun sistema que permite la gestión de contenido, enfocado en la creación de páginas webs.En la actualidad es una de las principales herramientas de software libre para la creación de páginas webs de todo tipo.
- **Alfresco:** es un sistema de administración de contenidos de código fuente libre, desarrollado en Java, basado en estándares abiertos y de escala empresarial. Está diseñado para usuarios que requieren un alto grado de modularidad y rendimiento escalable. Alfresco incluye un repositorio de contenidos, un marco de trabajo de portal web para administrar y usar contenido estándar en portales, una interfaz que provee compatibilidad de sistemas de archivos en Windows y sistemas operativos tipo Unix, un sistema de administración de contenido web, capacidad de virtualizar aplicaciones webs y sitios estáticos.
- **ProcessMaker:**es un sistema de gestión de procesos de negocio que permite a las organizaciones simplificar su flujo de trabajo a través de la captura y automatización de procesos de negocio, es desarrollado por la empresa Colosa Inc. bajo licenciamiento GNU AGPLv3[4].

El software ProcessMaker es la base tecnológica de la Plataforma “Bienestar” que en su integración con los demás sistemas permite la interacción entre los usuarios que se convierten en los actores de los diferentes procesos de negocios que en él se modelan.Esto lo convierte en el destino del componente a implementar.

El mecanismo que presenta dicho sistema para extender sus funcionalidadeses mediante el desarrollo de plugins[3], de esta forma permite agregar nuevas funcionalidades sin entorpecer el código propietario, permitiendo así que futuras actualizaciones no sobrescriban estas funcionalidades agregadas.

La arquitectura del ProcessMaker se abordará detalladamente en el siguiente capítulo.

1.4 Análisis de soluciones similares

A continuación, se presenta un análisis de los sistemas de comunicación sincrónica más utilizados en el mundo y en Cuba con el objetivo de conocer las tendencias actuales y tecnologías empleadas en el desarrollo de estos.

1.4.1 Sistemas de comunicación sincrónica a nivel internacional

1.4.1.1 Skype

Skype es un software propietario distribuido por Microsoft tras haber comprado la compañía homónima y que permite comunicaciones de texto, voz y video sobre Internet (VoIP). Esta aplicación también incluye una característica que permite a los usuarios llamar a teléfonos convencionales, cobrándoles diversas y bajas tarifas según el país de destino, pudiendo llamar a casi cualquier teléfono convencional del mundo[5].

Skype utiliza como protocolo MSNP24 que es un protocolo cerrado y privativo de mensajería instantánea creado por Microsoft para su propia red de mensajería. A pesar de que Skype cuenta con clientes para Windows, MacOS y Linux, el hecho de que el protocolo sea cerrado no permite el desarrollo de una aplicación cliente compatible con los servicios de Skype, lo que imposibilita la adaptación de este sistema para ser utilizado en otro producto.

1.4.1.2 Hangouts

Hangouts es una aplicación de mensajería multiplataforma desarrollada por Google. Se creó para sustituir los servicios Google Talk, Google+ Messenger y Google+ Hangouts, unificando todos estos servicios en una única aplicación.

Hangouts permite mantener conversaciones entre dos o más usuarios, también es posible realizar videollamadas con hasta 15 personas en web y 10 personas desde un teléfono inteligente. Las conversaciones realizadas se archivan en la nube permitiendo con esto sincronizarlas entre diferentes dispositivos[6].

[3] En español se puede traducir como complemento. Es una aplicación que añade una funcionalidad adicional o una nueva característica al software.

Hangouts utiliza un protocolo de comunicación propietario y cerradode desarrollo propio que imposibilita la creación de un nuevo cliente[7].

1.4.1.3 WhatsApp

WhatsApp es una aplicación de mensajería principalmente enfocada en teléfonos inteligentes, en la que se envían y reciben mensajes mediante Internet, complementando servicios de mensajería instantánea, servicio de mensajes cortos o sistema de mensajería multimedia. Además de utilizar la mensajería en modo texto, también permite realizar llamadas de audio y video[8].Según datos recientes, es líder en mensajería instantánea en gran parte del mundo, en el que supera los 2000 millones de usuarios[9].

Esta aplicación utiliza XMPP[4] como protocolo que a pesar de ser estándar y completamente abierto, WhatsApp está totalmente ligado a un número de teléfono celular y esto imposibilita su utilización desde una plataforma externa.

1.4.2 Sistemas de comunicación sincrónica a nivel nacional

1.4.2.1 Plataformaintegral de comunicación Platel

Platel es una plataforma integral de comunicaciones desarrollada por la XETID,está completamente basada en software libre. Ofrece servicios de voz, datos, video, mensajería unificada, seguridad, supervisión y notificaciones, usando protocolos estándares e interfaces flexibles, con la capacidad de inter-operar con otros sistemas empresariales.

Platel cuenta con una arquitectura de software segura, robusta y flexible, que permite configurar sus funciones y servicios de acuerdo con las características y necesidades de cada cliente, para optimizar la administración de los servicios de comunicación. Soporta una amplia gama de tecnologías IP, interfaces para la conexión a la red telefónica pública conmutada y a la internet. En su núcleo cuenta con elementos estándares de comunicación IP y no IP, sobre el cual evolucionan servicios de valor agregado que valorizan la solución y la adaptan a cualquier escenario[10].

Principales características:

- **Mensajería unificada:** pone a disposición del cliente todos los mensajes en un formato al que se puede acceder en cualquier momento de manera unificada y lógica, sin importar la vía de acceso al mensaje.
- **Voz sobre IP (VoIP):** Platel está equipado para soportar los principales códec de voz, y los protocolos más difundidos de Voz sobre IP.

[4] Extensible Messaging and Presence Protocol, más conocido como XMPP (Protocolo extensible de mensajería y comunicación de presencia)

- **Videoconferencia:** Platel ofrece servicio de videoconferencia completo. La videollamada ofrece una opción a las empresas para que efectúen procesos de trabajo de una forma más productiva, permitiendo el ahorro de tiempo y evitando gastos en desplazamiento.
- **Presencia integrada:** la plataforma brinda información sobre la disponibilidad de sus usuarios; ofrece alternativas adicionales para contactar a un usuario.

Platel utiliza el protocolo abierto SIP[5],desarrollado por la organización IETF[6]con la intención de ser el estándar para la iniciación, modificación y finalización de sesiones interactivas de usuario donde intervienen elementos multimedia como video, voz y mensajería instantánea. La sintaxis de sus operaciones se asemeja a las de HTTP y SMTP, los protocolos utilizados en los servicios de páginas webs y de distribución de correos electrónicos respectivamente. Esta similitud es natural ya que SIP fue diseñado para que la telefonía se vuelva un servicio más en Internet[11].

1.4.3 Resumen de los sistemas de comunicación sincrónica existentes

El análisis realizado a las herramientas de comunicación sincrónica permitió elaborar unresumen en forma de tabla teniendo en cuenta los siguientes parámetros:

- **Procedencia:** referente a la procedencia del producto. Puede tomar valores Nacional o Extranjero.
- **Mensajería:** referente a si el sistema permite enviar y recibir mensajes de texto.
- **Videollamada:** referente a si el sistema permite realizar y recibir llamadas de audio y video.
- **Protocolo:**referente al protocolo utilizado por la aplicación.
- **Web:** referente a si el sistemacuenta con una aplicación web o no.

Herramienta	Procedencia	Mensajería	Videollamada	Protocolo	Web
Skype	Extranjera	Sí	Sí	MSNP24	Sí
Hangouts	Extranjera	Sí	Sí	Hangouts	Sí
WhatsApp	Extranjera	Sí	Sí	XMPP	Sí[7]
Platel	Nacional	Sí	Sí	SIP	Sí

Tabla 1: resumen de sistemas similares

A partir del estudio realizado de sistemas homólogos se arriba a las siguientes impresiones:

[5] Session Initiation Protocol (Protocolo de inicio de sesión)
[6] Internet Engineering Task Force (IETF) (en español, Grupo de Trabajo de Ingeniería de Internet)
[7] Cuenta con un cliente Web completamente ligado a unnúmero deteléfono celular

- Los sistemas Skype y Hangouts son softwares privativos y utilizan protocolos cerrados que imposibilitan la adaptación de los mismos para el uso dentro de la plataforma.
- El sistema WhatsApp no puede ser utilizado sin estar completamente ligado a un número celular.
- El software Platel cumple con todos los requisitos necesarios y tiene la ventaja de ser desarrollado en la empresa XETID, lo que lo convierte en el ideal para ser utilizado en la solución de la presente investigación.

A partir de estas impresiones se realizó un estudio sobre el cliente web de la Plataforma "Platel"encontrándose las siguientes desventajas en su integración con la Plataforma "Bienestar":

- Su interfaz visual no esta está desarrollada con el objetivo de ser embebida en otra página web.
- Está desarrollada utilizando el marco de trabajo JavaScript Angular.js que el autor de esta investigación lo considera muy robusto para un sistema tan pequeño como el que se desea desarrollar. Y principalmente para ser embebido dentro de una plataforma tan grande como "Bienestar", lo que podría influir negativamente en su rendimiento.
- No permite directamente la integración con los usuarios de la Plataforma "Bienestar".

Por todo lo antes expuesto, se puede concluir que el cliente web de Platel no puede ser utilizado directamente como propuesta de solución, lo quehace necesario el desarrollo de un nuevo clienteque permita ser integrado a la Plataforma "Bienestar".

1.5 Herramientas y tecnologías a utilizar para el desarrollo de la solución

A continuación se hace una descripción de las herramientas, lenguajes, metodologías y tecnologías específicas empleadas para guiar el desarrollo del componente. Las mismas fueron definidas por la XETID para el desarrollo de la Plataforma "Bienestar".

1.5.1 Metodología de desarrollo de software

Una metodología de desarrollo de software es un marco para las actividades, acciones y tareas que se requieren para desarrollar un software de alta calidad. Dicho marco define el enfoque que se toma cuando se diseña el software y proporciona una interacción entre los usuarios, diseñadores y las herramientas en evolución[12].Para el desarrollo del componente se utiliza como metodologíaPRODESOFT en su versión 1.7, esta es elaborada por la XETID ydefine las siguientes fases de desarrollo[13]:

- **Modelación:** se define una visión preliminar de la problemática a resolver por medio de lainformatización, los antecedentes de las soluciones informatizadas similares del problema planteado u otras experiencias similares. Se reflejan las generalidades de la propuesta de solución, el tipo de solución a emplear. Además de las principales características técnicas que tendrá la solución definiendo con estos elementos un proyecto técnico.Se capturan las partes esenciales del sistema, donde se identifican los procesos denegocio fundamentales y seaceptan los requerimientos funcionales, obteniéndose lalínea base de la arquitectura y una estrategia de construcción de la aplicaciónaprobada por los implicados en el proyecto. El hito fundamental de esta fase es laliberación de la arquitectura de sistema, datos y despliegue.
- **Construcción:** se aclaran los requisitos restantes y se completa el desarrollo del sistema sobre una base estable de la arquitectura. Las fases anteriores solo dieron una arquitectura básica que es aquí refinada de manera incremental conforme se construye el producto. En esta fase todas las características, componentes, y requerimientos deben ser integrados, implementados, y probados en su totalidad, obteniendo una versión liberada del producto.
- **Explotación Experimental:** se convierte la versión liberada del producto en una solución estable, donde se eliminan los errores que surgen durante las pruebas y se obtiene una certificación funcional y de seguridad del producto. Esta fase solo se ejecuta cuando se tiene como cliente al Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR), por requerimientos que exige este órgano.
- **Despliegue:** se instala y configura el sistema para un ambiente de producción real, se capacita al personal que usará la aplicación y se continúa dando soporte durante la explotación del sistema, culminando de ser preciso con transferencias tecnológicas.

Figura 2: fases de la metodología de desarrollo PRODESOFT

1.5.2 Lenguaje de modelado

Para el desarrollo de la presente investigación se emplea Lenguaje Unificado de Modelado (UML por sus siglas en inglés) en su versión 2.5. UML es un lenguaje de modelado estandarizado que permite a los desarrolladores especificar, visualizar, construir y documentar artefactos de un sistema de software. Por lo tanto, hace que estos artefactos sean escalables, seguros y robustos en ejecución.Utiliza la notación gráfica para crear modelos visuales de sistemas de software[14].

1.5.3 Herramienta de modelado

Como herramienta de modelado de la solución se utilizó Visual Paradigm,esta es una herramienta de Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Es una potente herramienta multiplataforma de diseño y gestión fácil de usar para sistemas de Tecnologías de la Información. Facilita una excelente interoperabilidad con otras herramientas CASE y la mayoría de los Entornos de Desarrollo Integrado[15].Para el desarrollo de la presente investigación se emplea Visual Paradigm en su versión 15.1.

1.5.4 Herramienta para el control de versiones

En el desarrollo de la solución se utiliza Apache Subversion (SVN) en su versión 1.12 ya que es el utilizado por la XETID para el control de versiones de todos sus productos. SVN es una herramienta de control de versiones de código abierto basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros. Utiliza el concepto de revisión para guardar los cambios producidos en el repositorio. Entre dos revisiones solo guarda el conjunto de modificaciones, optimizando así al máximo el uso de espacio en disco. SVN permite al usuario crear, copiar y borrar carpetas con la misma flexibilidad con la que lo haría si estuviese en su disco duro local. Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras. A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración[16][17].

1.5.5 Lenguajes de programación

Un lenguaje de programación es un lenguaje formal que, mediante una serie de instrucciones, le permite a un programador escribir un conjunto de órdenes, acciones consecutivas, datos y algoritmos para, de esa forma, crear programas que controlen el comportamiento físico y lógico de una máquina[18].En el desarrollo de la solución se utilizan los siguientes lenguajes de programación:

1.5.5.1 PHP

Acrónimo recursivo en inglés de PHP Hypertext Preprocessor (preprocesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor. Originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML[19].

Para el desarrollo del componente se emplea PHP en la versión 5.6.

1.5.5.2 JavaScript

Consiste en un lenguaje de programación interpretado, que habitualmente se utiliza en sitios webs para ejecutar acciones del lado del cliente, estando embebido en el código fuente de la página web. Técnicamente, constituye un dialecto del estándar ECMAScript, propuesto por la entidad internacional de estándares de información y comunicación ECMA International y diseñado inicialmente por Netscape y, posteriormente, por la Fundación Mozilla. También constituye un estándar ISO. Debido a su propósito y uso general, todos los navegadores webs modernos interpretan correctamente JavaScript, siendo un lenguaje universal y multiplataforma[20][21].

Para el desarrollo del componente se utilizará JavaScript en su estándar ECMAScript 5.

1.5.6 Otros lenguajes

En el progreso de la solución se utilizan los siguientes lenguajes que, a pesar de no ser considerados lenguajes de programación, actualmente son imprescindibles en el desarrollo de una aplicación web.

1.5.6.1 HTML

HTML es la sigla de **HiperText Markup Language (Lenguaje de Marcación de Hipertexto)** es un lenguaje es se utiliza comúnmente para establecer la estructura y contenido de un sitio web, tanto de texto, objetos e imágenes. El lenguaje de HTML funciona por medio de "etiquetas" que describen la apariencia o función del texto enmarcado. Este lenguaje puede llegar a incluir un script o código que tenga incidencia en el comportamiento del navegador web del cliente[22][23].

1.5.6.2 CSS

Acrónimo recursivo en inglés de CSS Cascading Style Sheets (hojas de estilo en cascada) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas webs complejas. Permite colocar colores de fondo, fuentes y marginando al contenido en múltiples columnas, entre otras funcionalidades

relacionadas a la apariencia visual. Para el desarrollo del componente es utilizado CSS en su versión 3[24][23].

1.5.7 Bibliotecas

Una biblioteca (en ciencias informáticas) es un conjunto de recursos (algoritmos) prefabricados, que pueden ser utilizados por el programador para realizar determinadas operaciones. Las declaraciones de las funciones utilizadas enestas bibliotecas, junto con algunas macros y constantes predefinidas que facilitan su utilización, seagrupan en ficheros de nombres conocidos que suelen encontrarse con extensiones .lib, .js, .dll, entreotras[25]. En el desarrollo de la solución se utilizan las siguientes bibliotecas:

1.5.7.1 Janus.js

Define una API[8] en el lenguaje de programación JavaScriptque permite la interacción con un servidor WebRTC Janus para la comunicación en tiempo real mediante texto, audio y video. Todas las interacciones están basadas en mensajes JSON.

1.5.7.2 Vue.js

Vue.js es una biblioteca para construir interfaces de usuario y aplicaciones de una sola página. A diferencia de otras librerías monolíticas, Vue está diseñada desde cero para ser utilizada incrementalmente. La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes[26]. Destaca principalmente en su facilidad de aprendizaje, su eficiencia y el tamaño reducido de su código fuente.Para el desarrollo del componente se utiliza Vue.js en su versión 2.6.

1.5.7.3 Stomp.js

STOMP[9]proporciona un formato interoperable para que los clientes de STOMP puedan comunicarse con cualquier agente de mensajes de STOMP para proporcionar una interoperabilidad de mensajería fácil y generalizada.Es un protocolo muy simple y fácil de implementar. Stomp.js es un cliente JavaScript que permite la conexión con servidores STOMP desde una página web. En la solución propuesta se utiliza para recibir la información sobre la presencia de los usuarios en la plataforma.

1.5.8 Marco de trabajo

En el desarrollo del componente se utiliza Gulliver como marco de trabajo. Gulliver es desarrollado por la empresa Colosa Inc. para ser utilizado en el desarrollo del sistema ProcessMaker. Es un marco de código abierto de alta calidad para desarrollar aplicaciones webs y servicios websen el lenguaje

[8] **API:**Application Programming Interface (Interfaz de Programación de Aplicaciones): constituye un mecanismo que facilita, mediante llamadas a funciones, implementar interacción entre la aplicación que proporciona dichas funciones y otra aplicación.
[9] **STOMP**: Simple Text Orientated Messaging Protocol, en español protocolo simple de mensajes de texto orientados.

PHP, basado en el patrón Modelo-Vista-Controlador y utilizando las ventajas que ofrece PHP en la programación orientada a objetos[27].

1.5.9 Formato de intercambio de datos

JSON acrónimo de JavaScript Object Notation (Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Para los humanos es sencillo leerlo y escribirlo mientras que las máquinas pueden interpretarlo y generarlo fácilmente. Está basado en un subconjunto del Lenguaje de Programación JavaScript aunque hoy se considera un formato de lenguaje independiente[28].JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia. Por lo antes expuesto, se emplea el formato JSON para el intercambio de datos entre el servidor y el cliente.

1.5.10 Entorno de desarrollo integrado

Un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) es un programa informático compuesto por un conjunto de herramientas de programación, díganse un editor de código, un compilador, un depurador y un constructor de Interfaz Gráfica de Usuario. En el desarrollo de la solución se utiliza Php Storm en su versión 2019.3, este es un IDE desarrollado por la compañía JetBrains especializado en el desarrollo web en el lenguaje de programación PHP, provee un completamiento inteligente de código, detecta código duplicado, mezcla lenguajes (PHP, JavaScript, CSS, HTML), presenta navegación rápida y chequeo de errores al momento, facilidades para refactorización, entre otras que ayudan en el desarrollo de la aplicación. Es un IDE ligero, de fácil instalación, multiplataforma y con una versión limitada que se distribuye de forma gratuita[29].

1.6 Conclusiones

- El estudio de la Plataforma “Bienestar” identificó que el principal mecanismo para la extensión de sus funcionalidades es mediante plugins.
- El análisis de sistemas homólogos, permitió identificar las tendencias en el desarrollo de herramientas para la comunicación sincrónica, así comola necesidad de desarrollar una solución basada en las potencialidades que brinda la Plataforma de Comunicaciones Platel.
- El análisis de la Plataforma “Bienestar” identificó las tecnologías y herramientas utilizadas por la empresa XETID en su desarrollo y que serán empleadas en el presente trabajo.
- Se definió la metodología PRODESOFT elaborada en la XETID para guiar el proceso de desarrollo de la solución.

Capítulo 2: Modelación de la solución

2.1 Introducción

En el presente capítulose describe la propuesta de solución con el objetivo de lograr una visión general de lo que se desea informatizar.También se describen las principales características que debe presentar la propuesta de solución, especificando los requisitos funcionales y no funcionales que debe cumplir el componente a desarrollar. De igual forma se fundamentan los patrones de diseño empleados,finalmente se diseña el diagrama de despliegue.

2.2 Propuesta de solución

Con el objetivo de facilitar la comunicación sincrónica entre los usuarios de la Plataforma “Bienestar” y aprovechando su flexibilidaden la extensión de sus funcionalidades mediante plugins, se propone el desarrollo de un nuevo cliente para la Plataforma Platel utilizando tecnologías webs y que cuente con lassiguientes funcionalidades:

- **Presencia integrada**: brinda información sobre la disponibilidad de los usuarios en la plataforma.
- **Mensajería instantánea:**comunicación en tiempo real entre dos usuarios basada en texto.
- **Llamadas de audio:** comunicación en tiempo real entre dos usuarios basada en audio.
- **Llamadas de video:** comunicación en tiempo real entre dos usuarios basada en video yaudio.

2.2.1 Modelo conceptual

Un modelo conceptual, es una representación visual en forma de diagrama de las clases conceptualesu objetos del mundo real que son significativos en un dominio de interés[30]. A continuación, se presenta el modelo conceptual con los principales conceptos identificados:

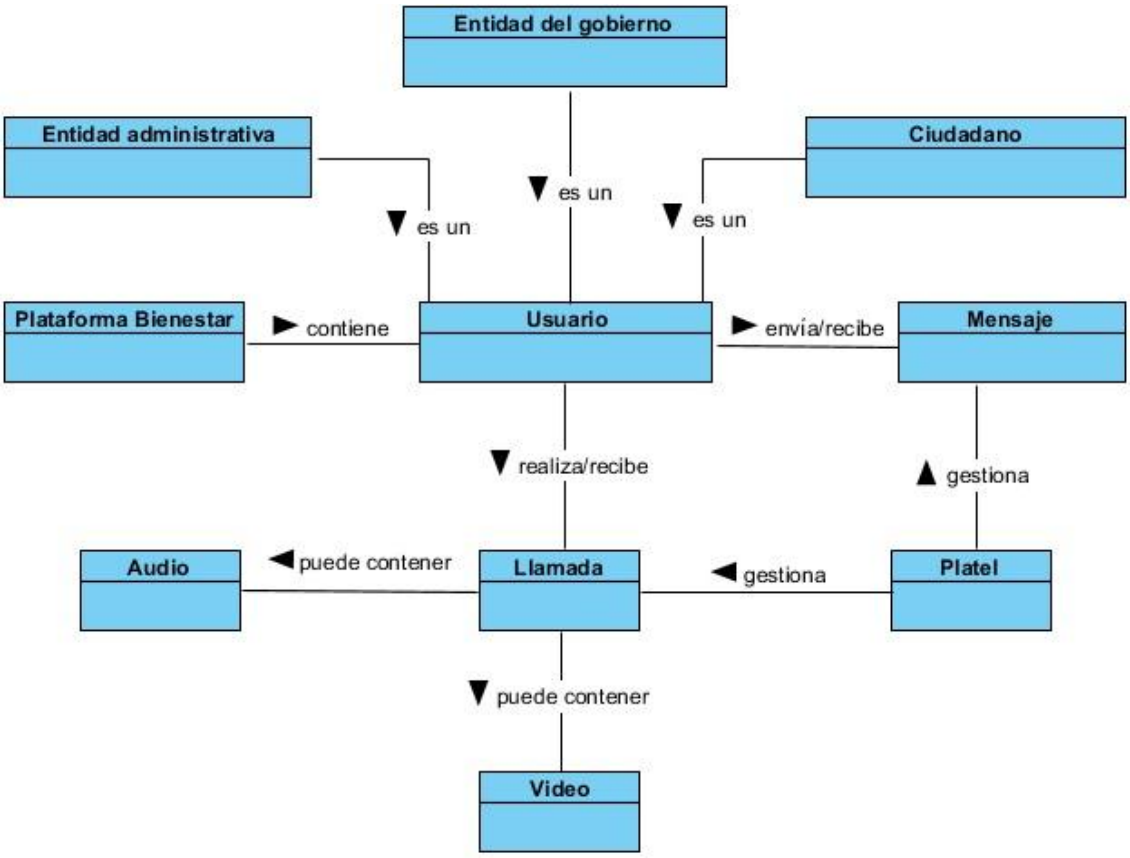


Figura 3: Modelo conceptual

Conceptos identificados:

- **Usuario:**es la persona que interactúa con el componente luego de autenticarse a la Plataforma “Bienestar”.
- **Entidad administrativa:**es un usuario que representa a una empresa que presta servicios a la población.
- **Entidad gubernamental:**usuario que representa a una entidad del gobierno referente a una zona administrativa del país.
- **Ciudadano:**usuario que utiliza la Plataforma “Bienestar” y no representan a ninguna institución estatal.
- **Plataforma “Bienestar”:**sistema dentro del cual interactúan los usuarios.
- **Platel:**plataforma de telecomunicaciones que gestiona la interacción entre los usuarios.
- **Mensaje:**mensaje de texto enviado o recibido por un usuario dentro de la plataforma.
- **Llamada:** operación en la que se comunican dos personas de la plataforma mediante audio y/o video.
- **Audio:** información que capta el micrófono del usuario.
- **Video:** información que capta la webcam del usuario.

2.3 Definición de requisitos

Los requisitos para un sistema son las descripciones de los servicios que un sistema debe proporcionar y las limitaciones de su funcionamiento. Estos requisitos reflejan las necesidades de los clientes de un sistema que sirve para un determinado propósito, como controlar un dispositivo, colocar un pedido o encontrar información[31].Los requerimientos deben ser especificados por escrito, claros y precisos. Estos se clasifican en:

- **Requisitos funcionales:** son tareas, capacidades o condiciones que el sistema debe cumplir. Dichas tareas dirigen un entendimiento de cuál será el impacto del software sobre el negocio, qué quieren los clientes y cómo los usuarios finales interactuarán con dicho software. Deben ser lo más completos, claros y precisos posibles[12].
- **Requisitos no funcionales:** definen cuáles son las propiedades con las que el sistema debe contar, así como sus restricciones fundamentales. Este tipo de requisitos garantizan que el producto sea fácil de usar, seguro y atractivo. Hacen referencia a las propiedades emergentes de un sistema como la fiabilidad, el tiempo de respuesta, el rendimiento y la seguridad[12].

2.3.1 Técnicas utilizadas para la obtención de requisitos

Para la obtención de los requisitos se utilizaron las siguientes técnicas:

- **Análisis de otros sistemas:** mediante esta técnica se pudieron estudiar sistemas de comunicación sincrónicasimilares al que se desea desarrollar. El estudio del funcionamiento, características, ventajas y desventajas de estos sistemasaportó en la identificaciónde los requisitos.
- **Entrevistas:** se realizaron entrevistas no estructuradas al equipo de desarrollo de la Plataforma “Bienestar”, con el objetivo de obtener toda la información posible sobre la visión que el entrevistado tiene sobre las funcionalidades del componente a desarrollar.

2.3.2 Identificación de requisitos

A continuación, se presentan los requisitos identificados a partir de las técnicas descritas anteriormente.

2.3.2.1 Requisitos funcionales

RF.01 Registrar usuario en la Plataforma “Platel”:permite el registro automático del usuario en la Plataforma “Platel”.

RF.02 Mostrar lista de usuarios: el componente debe permitir mostrar una lista con elnombre completo de todos los usuarios de la Plataforma “Bienestar”.

RF.03 Mostrar presencia de los usuarios: el componente debe mostrar si un usuario está conectado o no en el momento.

RF.04 Mostrar conversaciones:a partir de la selección de un usuario, el componente debe mostrar una ventana con las conversaciones que se hayan realizado con el mismo.

RF.05 Enviar mensaje: el componente debe permitir a un usuario enviar mensajes de texto en tiempo real a otro usuario conectado.

RF.06 Recibir mensaje: el componente debe permitir recibir mensajes de texto en tiempo real.

RF.07 Realizar llamada de audio: el componente debe permitir realizar llamadas de audio en tiempo real.

RF.08 Recibir llamada de audio: el componente debe permitir recibir llamadas de audio en tiempo real.

RF.09 Realizar videollamada: el componente debe permitir realizar videollamadas en tiempo real.

RF.10 Recibir videollamada: el componente debe permitir recibir videollamadas en tiempo real.

2.3.2.2 Requisitos no funcionales

Interfaz:

RnF.01 El componente debe aparecer en la Plataforma “Bienestar” en forma de una ventana flotante desplegable en la esquina inferior derecha.

RnF.02 Las llamadas de audio y video deben aparecer en una ventana independiente.

RnF.03 La aplicación debe funcionar sin que el usuario tenga que recargar la página donde esta se encuentre embebida.

Soporte:

RnF.04 El componente cumplirá con las normas de codificación y bibliotecas de clase definidas para la Plataforma “Bienestar”.

Software:

RnF.05 El cliente debe contar con uno de los siguientes navegadores:

- Mozilla Firefox versión 52 o posterior
- Google Chrome versión 73 o posterior

Seguridad y confiabilidad:

RnF.06 El componente debe garantizar la integridad y confidencialidad de la información que se transmite.

2.3.3 Especificación de requisitos funcionales

La especificación de los requisitos funcionales siguiendo lo definido por la metodología de desarrollo PRODEFSOFT incluye los conceptos tratados, precondiciones, post-condiciones, una descripción detallada del flujo del requisito, las validaciones y una propuesta de prototipos de interfaces de usuario. A continuación, se muestra la especificación del requisito funcional No.5 Enviar mensaje:

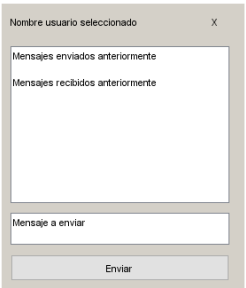
<table>
<tr><td rowspan="2">Conceptos</td><td>Conceptos</td><td>Atributos</td></tr>
<tr><td>Mensaje</td><td>destinatario, contenido</td></tr>
<tr><td rowspan="3">Precondiciones</td><td>Precondiciones</td><td>Pre-requisitos</td></tr>
<tr><td>El usuario debe estar autenticado.</td><td rowspan="2">Autenticar usuario en la Plataforma Platel.Mostrar lista de usuarios.
Mostrar conversaciones.</td></tr>
<tr><td>El usuario debe tener acceso a la funcionalidad.</td></tr>
<tr><td>Descripción</td><td colspan="2">1- El usuario se autentica correctamente en la Plataforma “Bienestar”.
2- El sistema muestra una ventana en la esquina inferior derecha con la lista de usuarios, sus fotos y si está conectado o no en ese momento.
3- El usuario selecciona a un usuario de la lista de usuarios disponible.
4- El sistema muestra la interfaz mostrar conversaciones con el usuario seleccionado.
5- El usuario escribe el texto a enviar en el campo correspondiente.
6- El usuario presiona el botón Enviar o presiona la tecla Enter.
7- El sistema envía el mensaje al usuario correspondiente.
8- El sistema muestra el nuevo mensaje enviado al final de la lista de mensajes.</td></tr>
<tr><td>Complejidad</td><td colspan="2">Crítico</td></tr>
<tr><td>Validaciones</td><td colspan="2">Ninguna</td></tr>
<tr><td>Post-condiciones</td><td colspan="2">Se ha enviado el mensaje.</td></tr>
<tr><td>Post-requisitos</td><td colspan="2">Ninguno</td></tr>
<tr><td>Prototipo de interfaz de usuario</td><td colspan="2"> 
</td></tr>
</table>

Tabla 2: requisito funcional No.5 Enviar mensaje

Las especificaciones del resto de los requisitos aparecen en el Anexo No.1.

2.3.4 Validación de requisitos

En la validación de los requisitos se contó con la participación del cliente y se comprobó que estos cumplen con las siguientes características definidas en PRODESOFT:

- Se definen todos los requisitos deseados.
- Cada requisito solo tiene una interpretación posible (se eliminaron ambigüedades).
- El cumplimiento de cualquier requisito no provoca conflictos con el cumplimiento de otro requisito.

2.4 Diseño de la arquitectura de sistema

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos [12]. En este caso, el componente que se desarrolla emplea la arquitectura del BPMS ProcessMaker. Dicha arquitectura se presenta en la figura siguiente:

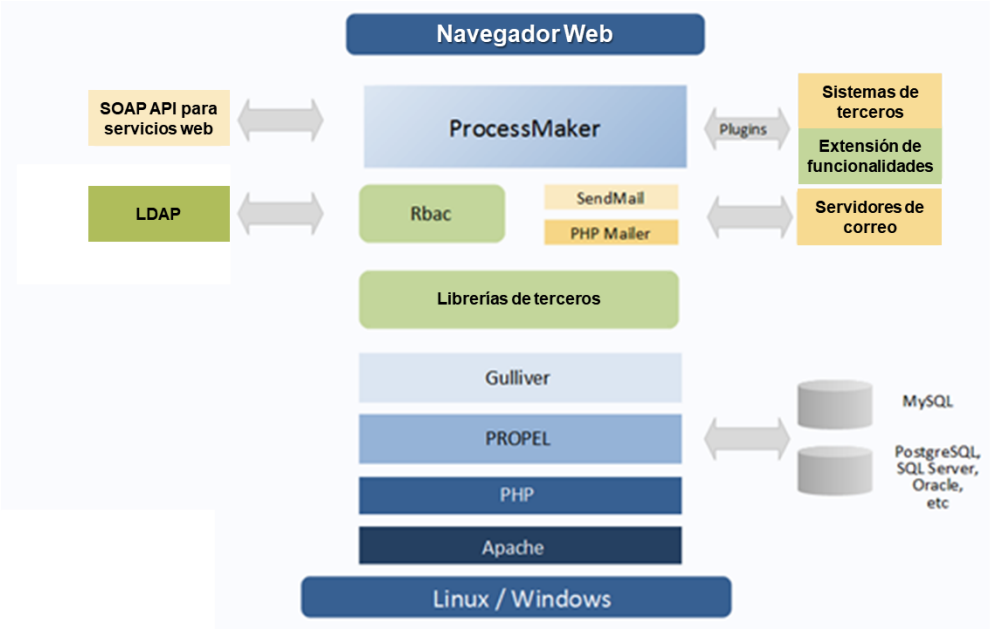


Figura 4: arquitectura de ProcessMaker[32]

Los componentes de la arquitectura del ProcessMaker son[32]**:**

- Usa el software de mapeo relacional de objetos (ORM) **Propel** para mapear clases y bases de datos, el mismo es de código abiertoescrito en **PHP**.
- Está diseñado en el marco de trabajo de **Gulliver** y usa los lenguajes de plantillas **Smarty**[10].

[10] Es un motor de plantillas para PHP. Se encuentra bajo la Licencia Pública General Reducida de GNU.

- Utiliza **RBAC**[11] para administrar roles de usuario y usando **LDAP**[12]es capaz de administrar usuarios de alto nivel de autenticación.
- Usa **WSO2** para administrar servicios webs con **SOAP**[13]y así poder interconectar fácilmente con otras aplicaciones como ERP[14], CMS[15] entre otros softwares de colaboración.
- Maneja 2 motores de correo: **SendMail** y **PHP Mailer**.
- Es fácil de implementar: 100% basado en la web, aunque está optimizado para Mozilla Firefox.
- Está escrito en el lenguaje **PHP** y utiliza como servidor web **Apache**.
- Permite la extensión de funcionalidades y la integración con softwares de terceros mediante el desarrollo de **plugins**.

2.4.1 Patrones arquitectónicos

Un patrón impone una regla a la arquitectura, describe la maneraen la que el software manejará ciertos aspectos de su funcionalidad en el nivel de la infraestructura. Proporciona un conjunto de sub-sistemas predefinidos, especificando sus responsabilidades, reglas, directrices que determinan la organización, comunicación, interacción y relaciones entre ellos[12].

En el caso delcomponente desarrollado se utiliza el patrón arquitectónico **MVC**[16] utilizado por el marco de trabajo Gulliver,este se basa en separar las funcionalidades del sistema en tres componentes: modelo, vista, y controlador. A continuación, se muestra el patrón arquitectónico MVC:

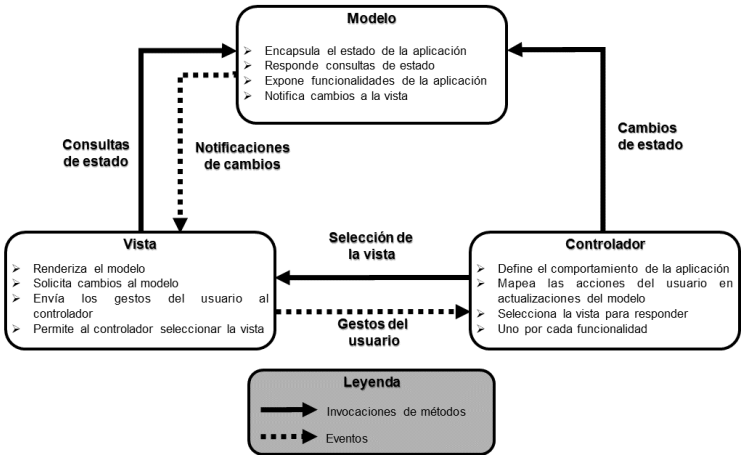


Figura 5: patrón arquitectónico Modelo-Vista-Controlador[33]

[11] Role Based Access Control por sus siglas en inglés, en español Control de Acceso Basado en Roles.
[12] Lightweight Directory Access Protocol por sus siglas en inglés, en español Protocolo Ligero/Simplificado de Acceso a Directorios.
[13] Simple Object Access Protocol por sus siglas en inglés, en español Protocolo de Acceso de Objetos Simples.
[14]Enterprise Resource Planning por sus siglas en inglés, en español Sistemas de Planificación de Recursos Empresariales.
[15]Content Management System por sus siglas en inglés, en español Sistema de Gestión de Contenidos.
[16]Modelo-Vista-Controlador.

Modelo:es una representación de los datos o el estado de la aplicación, contiene (o provee una interfaz) para la lógica de la aplicación[33].

Vista: es un componente de interfaz de usuario que produce una representación del modelo para el usuario y permite a este la entrada de datos[33].

Controlador: maneja la interacción entre el modelo y la vista, convirtiendo las acciones del usuario en cambios en el modelo o la vista[33].

2.4.2 Patrones de diseño

Los patrones de diseño son una solución reutilizable de problemas recurrentes que ocurren durante el desarrollo del software[34]. Para el diseño de la solución se hizo uso de los Patrones Generales de Software para Asignar Responsabilidades **(GRASP)**, de los patrones Gang of Four o Pandilla de los Cuatro **(GoF)**, así como de otros patrones de usocomún en el lenguaje de programación JavaScript.

2.4.2.1 Patrones GRASP

Los patrones de asignación de responsabilidades describen los principios fundamentales del diseño de objetos para la asignación de responsabilidades[34]. En el desarrollo de la solución se utilizaron los siguientes patrones GRASP:

- **Alta cohesión:** en cada clase solo se implementaron las funcionalidades que le corresponden.
- **Bajo acoplamiento:**cada clase implementada solo se comunica con un número relativamente pequeño de clases.
- **Creador:** las clases implementadas que tienen la responsabilidad de crear objetos contienen toda la información necesaria para construir los mismos.
- **Experto:** cada clase implementada mantiene el encapsulamiento, los objetos utilizan su propia información para llevar a cabo sus tareas.

2.4.2.2 Patrones GOF

La línea base en el tema de patrones de diseño la impone el catálogo "Design Patterns: Elements ofReusable Object-Oriented Software", en este libro se presenta un conjunto de 23 patrones de diseñoidentificados a partir del estudio y la experiencia del grupo llamado Banda de los Cuatro o The Gang ofFour (GOF): Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, quienes se dedicaron aanalizar los problemas recurrentes en el desarrollo de software y realizaron una clasificación y agrupacióna partir de dos criterios, su propósito y alcance, las categorías definidas según el propósito son[35]:

- **Patrones Creacionales:** inicialización y configuración de objetos.

- **Patrones Estructurales:** separan la interfaz de la implementación. Se ocupan de cómo las clases yobjetos se agrupan, para formar estructuras más grandes.
- **Patrones de Comportamiento:** más que describir objetos o clases, describen la comunicación entreellos.

A partir de las categorías mencionadas se describen los patrones **GOF** empleados en el desarrollo delsoftware:

Patrón Singleton (Instancia única):

Es un patrón de diseño de tipo creacional, que permite restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella [34].

La propia clase es responsable de crear la única instancia y permite el acceso global a dicha instancia, declarando el constructor de clase como privado para que no sea instanciable directamente.

En el desarrollo de la solución se utiliza para garantizar que solamente se cree una instancia de la clase StompClient, responsable de obtener los usuarios que están conectados al componente.

```
let StompClient = function () {
    let instance = null;

    let isSubscribed = false;
    ...
    function unHexlify(s) {...}
    ...
    function create(){...}

    return{
        getInstance: function () {
            if(instance == null){
                instance = create();
            }
            return instance;
        }
    }
}();
```

Figura 6: Patrón instancia única

Patrón Facade (Fachada):

Es un patrón de diseño de tipo estructural.Proporciona una interfaz unificada a un conjunto de interfaces en un subsistema. Fachada define una interfaz de nivel superior que hace que el subsistema sea más fácil de usar.En el desarrollo de la solución la clase WebPhone constituye una implementación del patrón Fachada ya que proporciona una interfaz unificada para la utilización de las clases Janus y Stomp, de esta forma todas las operaciones relacionadas con la interacción con Platel se encuentran unificadas en una sola clase[34].

```
connect: function () {
    createStompClient();
    initJanusAndConnect();
},
disconnect: function () {
    unregister();
    stompClient.disconnect();
},
registerRefresh: function () {
    register( refresh: true);
},
sendMessage: function (to, content) {
    let transactionId = Janus.randomString(12);
    let date = Date.now();
    reSendMessage(to, content);
},
```

Figura 7: Patrón fachada

Patrón Observer(Observador):

Es un patrón de diseño de tipo comportamiento que define una dependencia de uno a muchos entreobjetos de forma tal que cuando un objeto cambia su estado todas sus dependencias son notificadas y actualizadas automáticamente[34].

En el desarrollo de la solución se utiliza en la clase WebPhone para notificar sobre los eventos que se generan al recibir mensajes, llamadas u otros.

Figura 8: Patrón observer

Figura 9: Patrón observer

```
...
} else if (event === 'hangup') {
    sipCall.hangup();
    events.publish( topic: "hangup");
} else if (event === 'message') {
...
```

La Figura 9 muestra cómo sepublica el evento hangup en la clase WebPhone al recibir la información de que la llamada en curso ha terminado. La Figura 8 muestra la suscripción al evento por parte de la clase Main.

2.4.2.3 Otros patrones

Patrón Módulo:

El patrón módulo se definió originalmente como una forma de proporcionar encapsulación pública y privada para las clases de ingeniería de software convencional. En JavaScript, el patrón módulo se utiliza para emular aún más el concepto de clases de tal manera que podamos incluir tanto métodos públicos y privados como variables dentro de un solo objeto, protegiendo así partes particulares del ámbito global. Esto resulta en una reducción en la probabilidad de que los nombres de funciones entren en conflicto con otras funciones definidas en scripts adicionales en la página [36].

El patrón del módulo encapsula privacidad, estado yorganización mediante closures[17]. Esoproporciona una forma de envolver una combinación de métodos y variables públicas y privadas, evitando que las piezas se filtren en el ámbito global y colisionen accidentalmente con la interfaz de otro desarrollador.

Con este patrón, solo se devuelve una API pública, manteniendo todo lo demás dentro del closure privado[36].

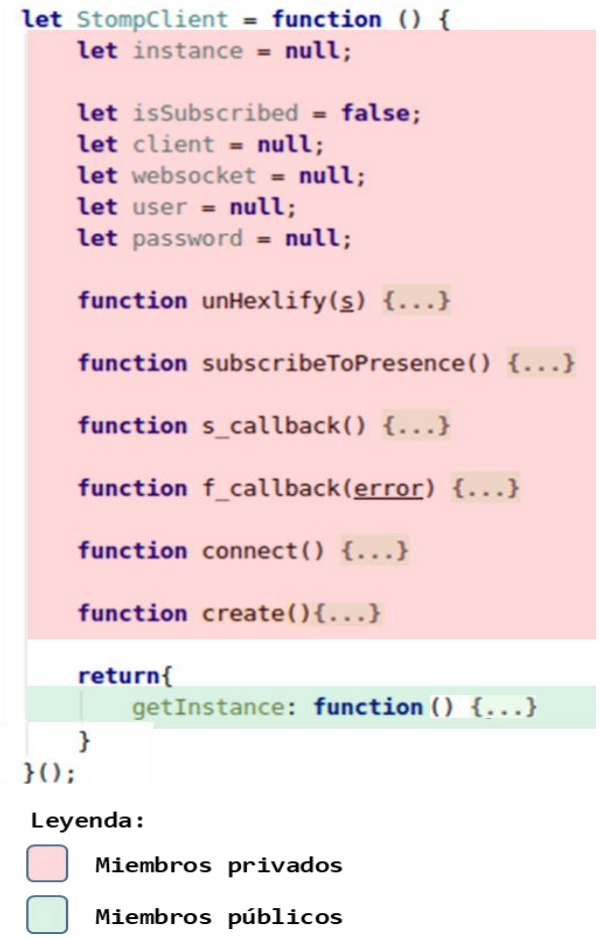


Figura 10: Patrón modulo

En el desarrollo de la aplicación se hace uso de este patrón con el objetivo de simular en JavaScript la encapsulación de los lenguajes orientados a objetos tradicionales.

En la Figura 10, se muestra un ejemplo de utilización del patrón en la clase StompClient donde los atributos y funciones dentro del recuadro rojo son privados y solo son accesibles dentro de la clase StompClient mientras quela funcióngetInstance() es pública y permite ser accedida desde afuera de la clase.

[17]Un closure en español clausura o cerradura es una técnica para implementar ámbitos léxicos en un lenguaje de programación con funciones de primera clase. Es un registro que contiene una función junto con el ámbito donde fue declarada.

2.4.3 Diagrama de componentes

Un componente representa una parte de un sistema modular, desplegable, y reemplazable, que encapsula la implementación y expone un conjunto de interfaces [12]. Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. A continuación, se presenta el diagrama que muestra la dependencia entre los componentes que intervienen en la solución:

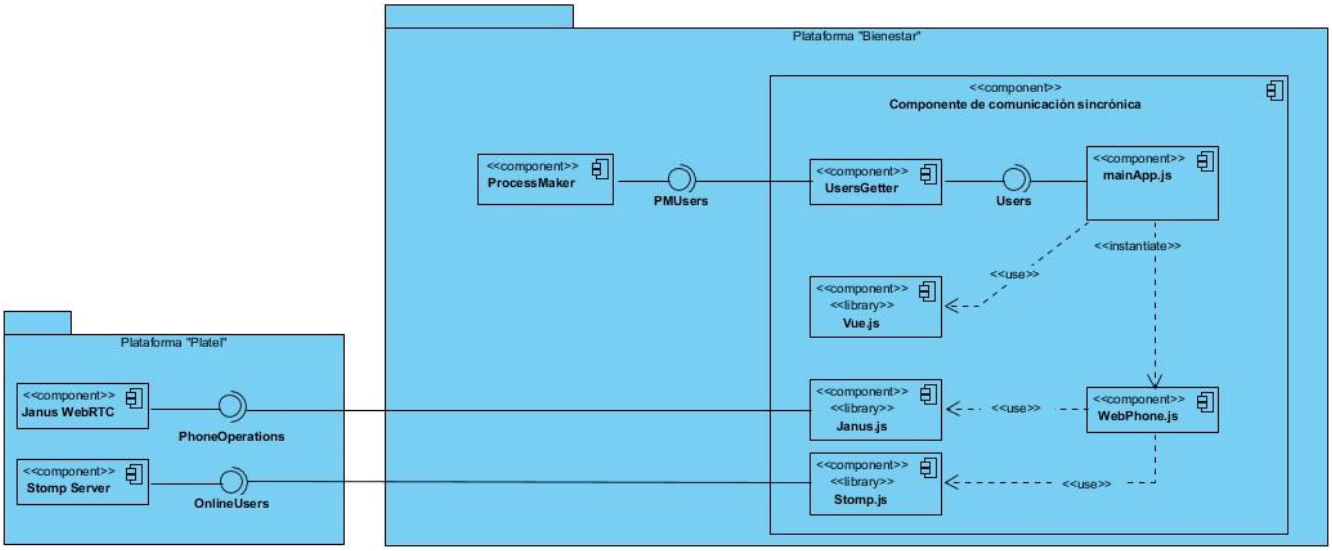
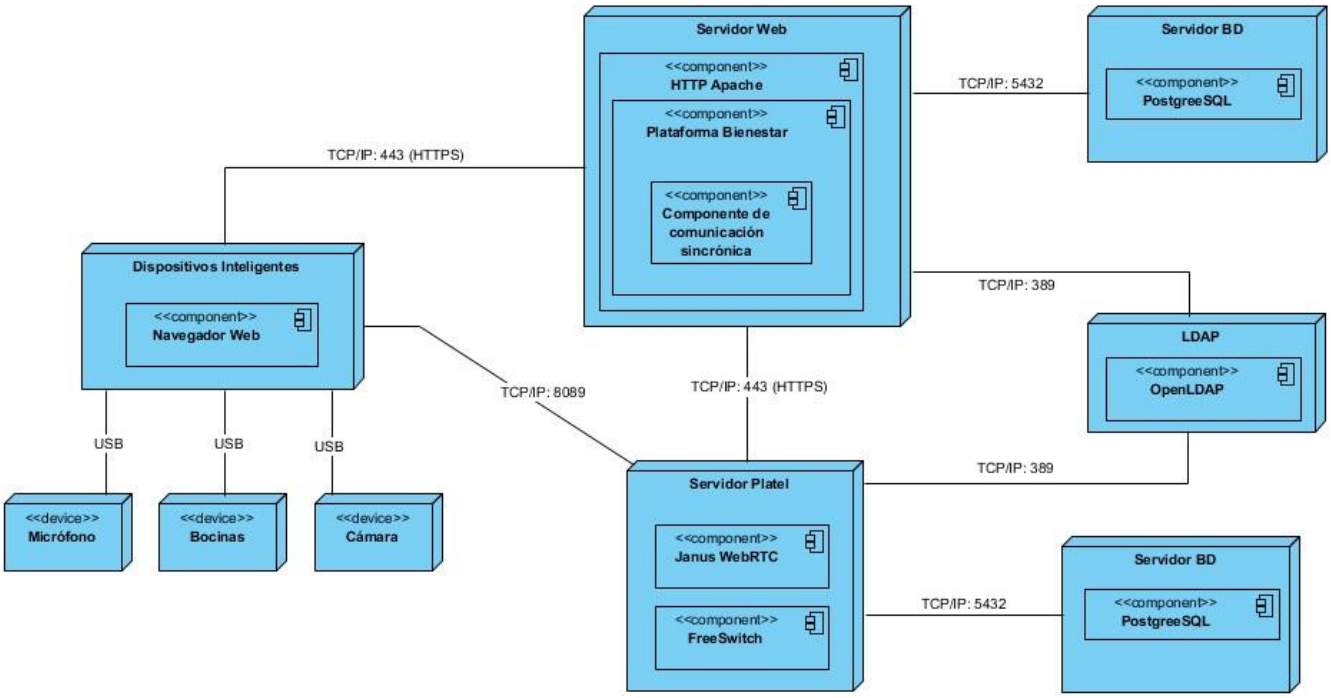


Figura 11: diagrama de componentes

2.4.4 Modelo de despliegue

Un modelo de despliegue permite indicar cómo se ubicarán las funcionalidades y los subsistemas dentro del entorno computacional físico que soportará el software. Muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los vínculos de comunicación entre ellos y las instancias de los componentes[12]. A continuación, se muestra el modelo de despliegue diseñado para la aplicación:



2.5 Conclusiones

Después de realizada la fase de modelación de la metodología PRODESOFT se arriba a las siguientes conclusiones:

- El modelo conceptual de la propuesta de solución permitió representar los conceptos significativos del negocio y la relación que existe entre estos.
- A partir de las funcionalidadesbrindadas por la Plataforma Platel y las entrevistas realizadas con el cliente se identificaron 10requisitos funcionales y 6 requisitos no funcionales.
- La realización del diagrama de componentes permitió mostrar la interacción del componente con las plataformas Platel y “Bienestar”, así como lasdependenciasde librerías externas.
- La realización del diagrama de despliegue permitió conocer las relaciones físicas entre el cliente y los nodos de las plataformas Platel y “Bienestar”.
- Se emplea el patrón arquitectónico modelo-vista-controladorutilizado en el desarrollo de la Plataforma “Bienestar” y se describen 4 patrones de diseño utilizados durante el desarrollo.

Capítulo 3: Construcción de la solución

En este capítulo se describe la implementación de la solución, donde se materializa el producto final y se cumple con los requisitos especificados en el capítulo anterior. Se diseñan los prototipos de interfaces funcionales y diagramas de clases. Se definen reglas o estándares de codificación para una mayor uniformidad y entendimiento del código fuente. A partir del código resultante y su funcionamiento se diseñan y ejecutan las pruebas desoftware a la solución desarrollada. Finalmente se valida la investigación utilizando el método de criterio de expertos.

3.1 Implementación de la solución

Los objetivos de esta fase van destinados a desarrollar de forma iterativa e incremental un producto completo.

3.1.1 Estándares de codificación

La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. La mantenibilidad del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento. Aunque la legibilidad y la mantenibilidad son el resultado de muchos factores, una faceta del desarrollo de software en la que todos los programadores influyen especialmente es en la técnica de codificación. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego revisiones del código. Para el desarrollo de la solución seemplea el siguienteestándar de codificación utilizado por XETID en la Plataforma “Bienestar”:

3.1.1.1 Indentación[18], llaves y tamaño de líneas

Usar una indentación sin tabulaciones, con un equivalente a 4 espacios. La apertura de llaves “{” seráen la misma de definición de la función. La longitud de las líneas de código es de un máximo de 120 caracteres.

3.1.1.2 Convención de nomenclatura

Variables: se rigen por la nomenclatura camelCase [19] . Siempre comienzan con minúscula (lowerCamelCase) y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula excepto la primera.

[18] Es un anglicismo de la palabra inglesa indentation, de uso común en la informática. No es un término reconocido por la Real Academia Española. En los lenguajes de programación es un tipo de notación secundaria utilizado para mejorar la legibilidad del código fuente por parte de los programadores.

[19] Es un estilo de escritura comúnmente utilizado en programación que se aplica a frases o palabras compuestas.

Clases:siempre comienzan con mayúscula (UpperCamelCase) y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula.

Funciones: siempre comienzan con minúscula(lowerCamelCase) y en caso de nombres compuestos la primera letra de cada palabra comienza conmayúscula excepto la primera. Los parámetros son separados por espacio luego de la coma.

Ficheros: todo siempre en minúscula y en caso de nombres compuestos se usa el guion bajo.

3.1.1.3 Estructuras de control

Las estructuras de control incluyen if, for, foreach, while, switch. Entre las estructuras de control y los paréntesis debe de existir un espacio. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales. Con esto se aumenta la legibilidad del código y se disminuye la probabilidad de errores lógicos.

3.1.1.4 Espacios en blanco

Se deben usar dos líneas en blanco entre las diferentes secciones de un fichero de código fuente. Una línea en blanco entre métodos, variables locales de un método y la primera sentencia, antes de un bloque o un comentario de línea, y entre secciones lógicas dentro de un fichero.

3.2 Pruebas de software

Las pruebas del software son la actividad más común de control de la calidad realizada en los proyectos para asegurar el correcto funcionamiento del software. Tienen como objetivos la verificación de la correcta implementación de los requisitos explícitamente establecidos, la adecuada integración de los componentes que conforman el sistema y la ejecución de casos de prueba que permitan detectar el mayor número de no conformidades y corregirlas antes de la entrega del software al cliente[13].

Es importante destacar que las pruebas reducen la probabilidad de que aparezcan defectos ocultos en el software, pero incluso si no se encuentra ningún defecto, nunca será una garantía de su corrección[13].

En el desarrollo de la solución se utilizan los siguientes métodos de prueba:

Método de caja blanca:

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de

componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, se pueden derivar casos de prueba que[12]:

- Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.
- Revisen todas las decisiones lógicas en sus lados verdadero y falso.
- Ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas.
- Revisen estructuras de datos internas para garantizar su validez.

Método de caja negra:

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en losrequerimientos funcionales del software; es decir, las técnicas de prueba de caja negra permitenderivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para lastécnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable quedescubra una clase de errores diferentes que los métodos de caja blanca[12].

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

- Funcionalidades incorrectas o faltantes.
- Errores de interfaz.
- Errores en las estructuras de datos o en elacceso a bases de datos externas.
- Errores de comportamiento o rendimiento
- Errores deinicialización y terminación.

3.2.1 Estrategia de pruebas

Para evaluar la calidad del componente que se está desarrollando y verificar el cumplimiento de los objetivos trazados, se diseña la siguiente estrategia de prueba para aplicar a la solución desarrollada:

Prueba	Método	Técnica
Prueba de unidad	Caja blanca	Camino básico
Prueba de integración	Caja negra	Incremental
Prueba de validación	Caja negra	Prueba basada en modelo
Prueba de sistema: Prueba de recuperación Prueba de seguridad Prueba de rendimiento	Caja negra	Semiautomático

Tabla 3: pruebas aplicadas al componente

3.2.2 Pruebas de unidad

La prueba de unidad enfoca los esfuerzos de verificación en la unidad más pequeña del diseño de software: el componente o módulo de software. Al usar la descripción del diseño de componente como guía, las rutas de control importantes se prueban para descubrir errores dentro de la frontera del módulo. La relativa complejidad de las pruebas y los errores que descubren están limitados por el ámbito restringido que se establece para la prueba de unidad. Las pruebas de unidad se enfocan en la lógica de procesamiento interno y de las estructuras de datos dentro de las fronterasde uncomponente[12].

Para la realización de la prueba se utiliza la técnica de camino básico, la cual permite obtener una medida de la complejidad lógica del diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución, garantizando con estos que durante la prueba se ejecute por lo menos una vez cada sentencia del programa [12].

Esta prueba se realiza de forma automática en el sistema, para ello se utiliza la librería PHPUnit. La librería permite determinar con certeza si un código específico produce el tipo de dato y resultado esperado.

A continuación, se muestra el caso de prueba del método getAllUsers() perteneciente a la funcionalidad mostrar lista de usuarios,como resultado la funcionalidad se espera obtener una lista con todos los usuarios existentes en el sistema y se establece como convención que el primer usuario de la lista será el usuario que actualmente se encuentra autenticado.

Prueba estructural de caja blanca	**Código de caso de prueba**
	getAllUsers()
Probador	Rolando Ramírez Hidalgo
Tipo de dato esperado	Lista de usuarios

Código al que se aplica:

```
public function getAllUsers() {
    $res = [];

1   $loggedUsername = pmPlatel_getLoggedUser()["USR_USERNAME"];
    $pmUsers = pmPlatel_getAllUsers();

2   foreach ($pmUsers as $pmUser) {
3       $user = $this->makeUser($pmUser);

4       if ($user->userName === $loggedUsername) {
5           array_unshift( &array: $res, $user);
        } else {
6           $res[] = $user;
        }
    }

7   return $res;
}
```

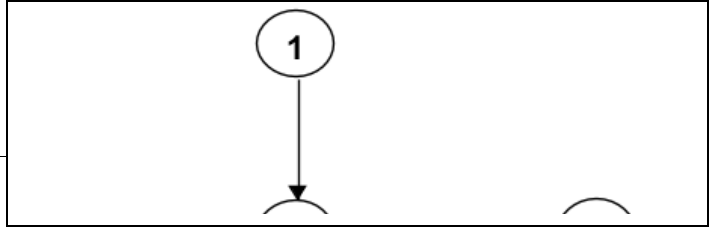
Complejidad ciclomática:

V(G) = E - N + 2

V(G) = 8 – 7 + 2 = 3

Caminos independientes:

Representación del grafo:



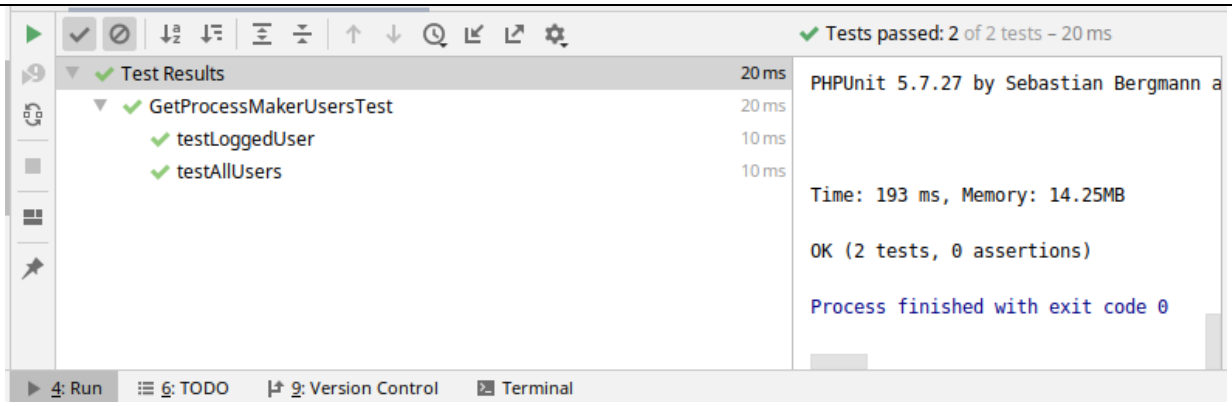
1) 1, 2, 7 2) 1,2,3, 4, 5, 2, 7 3) 1, 2, 3, 4, 6,2, 7	
Caso de prueba para los caminos básicos:	
Tipo de dato esperado:	Vacío
Funciones de evaluación:	

```php
function testLoggedUser(){
    $users = getAllUsers();
    assert( assertion: $users[0]->userName == 'rolando');
}


function testAllUsers(){
    $users = getAllUsers();
    assert( assertion: count($users) == 4);

    assert( assertion: $users[1]->userName == 'administrador');
    assert( assertion: $users[2]->userName == 'anonimo');
    assert( assertion: $users[3]->userName == 'platel');
}
```

Resultados de la prueba:



Evaluación dela prueba:	Satisfactoria

En el caso del camino independiente No.1 es imposible que ocurra, ya que implicaría la no existencia de usuarios en la base de datos del sistema. Las dos pruebas ejecutadas cubren los caminos independientes 2 y 3.

Las pruebas unitarias fueron aplicadas a lo largo del proceso de codificación del componente, corrigiéndose correctamente todos los errores encontrados en el momento de aparición.

3.2.3 Pruebas de integración

Las pruebas de integración son una técnica sistemática para construir la arquitectura del software y, al mismo tiempo, realizar pruebas para descubrir errores asociados con la interconexión. El objetivo es tomar componentes probados por unidades y construir la estructura del programa que se diseñó[12].

En el desarrollo de esta prueba se utiliza la técnica de integración incremental. El programa se construye y se prueba en pequeños incrementos, donde los errores son más fáciles de aislar y corregir. Es más probable que las interfaces se prueben completamente y se puede aplicar un enfoque de prueba sistemático [12]. Se emplea el método de integración incremental por las facilidades que brinda al desarrollador de realizar pruebas al programa menos complejas y obteniendo mejores resultados.

Dentro de la técnica de integración incremental se utilizó la estrategia de integración top-down (descendente).La prueba de integración descendente es un enfoque incremental para la construcción de la arquitectura del software. Los módulos se integran moviéndose hacia abajo a través de la jerarquía de control, comenzando con el módulo de control principal (programa principal). Los módulos subordinados al módulo de control principal se incorporan a la estructura ya sea en

profundidad o primero en amplitud. El proceso de integración se realiza en una serie de cinco pasos[12]:

1. El módulo de control principal se utiliza como controlador de prueba y los apéndices se sustituyen por todos los componentes directamente subordinados al módulo de control principal.
2. Dependiendo del enfoque de integración seleccionado (es decir, primero en profundidad o primero en amplitud), los apéndices subordinados se reemplazan uno a la vez con componentes reales.
3. Las pruebas se realizan a medida que se integra cada componente.
4. Al completar cada conjunto de pruebas, se reemplaza otro apéndice con el componente real.
5. Se pueden realizar pruebas de regresión[20] para garantizar que no se hayan introducido nuevos errores.

El proceso continúa desde el paso 2 hasta que se construye toda la estructura del programa.

3.2.3.1 Detalles de la prueba

La prueba se encuentra dividida en tres fases.

1- **Interacción de usuario:**en esta fase se revisan los datos de entrada y salida, el formato de presentación de los datos, el procesamiento de errores y la presentación de los mismos.

2- **Comunicación:**en esta fase se revisa la capacidad del componente para comunicarse con la Plataforma “Platel”.

3- **Procesamiento de eventos:** en esta fase se revisan las acciones del componente en el procesamiento de los eventos.

3.2.3.2 Ejemplo de prueba de integración

Se aplica la prueba de integración al componente “Mostrar lista de usuarios”, obteniéndose los resultados que a continuación se muestran.

Sistema/componente al que se integra	ProcessMaker
Condiciones deejecución	Existe al menos un usuario registrado en el ProcessMaker.

[20]Es la nueva ejecución de algúnsubconjunto de pruebas que ya se realizaron a fin de asegurar que los cambios no propagaronefectos colaterales no deseados.

Descripción delaprueba	Comprobar que el componente “Mostrar lista de usuarios” proporcione la información referente a los usuarios registrados en el ProcessMaker.
Pasos de ejecución	El probador interactúa con el componente. El probador compruebaque el componente muestre todos los usuarios existentes en la base de datos, con su correspondiente información, así como sus fotos de forma correcta. Verificar que el usuario que se utiliza para la realización de la prueba no aparezca en la lista. Y que aparezca en la cabecera de la ventana como el usuario conectado.
Resultados esperados	El componente “Mostrar lista de usuarios” brinda toda la información solicitada.
Evaluación	Prueba satisfactoria

Tabla 4: caso de prueba de integración del componente“Mostrar lista de usuarios”

3.2.4 Pruebas de validación

Las pruebas de validación comienzan en la culminación de las pruebas de integración, cuando se ejercitaron componentes individuales, el software está completamente ensamblado como un paquete y los errores de interfaz se descubrieron y corrigieron. Las pruebas se enfocan en las acciones visibles para el usuario y las salidas del sistema reconocibles por el usuario. La validación puede definirse en muchas formas, pero una definición simple (aunque dura) es que la validación es exitosa cuando el software funciona en una forma que cumpla con las expectativas razonables del cliente [12].

Para el desarrollo de las pruebas de validación se utilizó la técnica de prueba basada en modelo (PBM), esta es una técnica de prueba de caja negra que usa la información contenida en el modelo de requerimientos como la base para la generación de casos de prueba.La PBM ayuda a descubrir errores en el comportamiento del software y, como consecuencia, es extremadamente útil cuando se prueban aplicaciones impulsadas por eventos [12].

Esta prueba se desarrolló de forma manualutilizando dos usuarios de pruebas autenticados desde dos dispositivos equipados con cámaras y videos. Esto permitió a partir de las especificaciones de los requisitos, comprobar el correcto cumplimiento de cada de uno ellos con el sistema funcionando de la forma más parecida posible al funcionamiento en producción.A continuación, se muestra la descripción del caso de prueba de validación para el requisito funcional No. 5 Enviar mensaje:

Condiciones de ejecución	El usuario está autenticado al sistema y tiene acceso a la funcionalidad.

Descripción de la prueba	Comprobar que el sistema envié el mensaje correctamente.
Pasos de ejecución	1. Los usuarios probador1 y probador2 se autentican en el sistema desde dos dispositivos diferentes. 2. El usuario probador1 selecciona al usuario probador2 en la lista de usuarios. 3. El usuario probador1 envía un mensaje al usuario probador2. 4. El usuario probador2 comprueba que el usuario probador1 aparezca parpadeando en la lista de usuarios. 1. El usuario probador2 selecciona al usuario probador1 en la lista de usuarios y verifica que el mensaje enviado por probador1 aparezca correctamente.
Resultados esperados	El usuario probador2 recibe el mensaje correctamente.
Evaluación	Prueba satisfactoria

Tabla 5: caso de prueba de validación al requisito funcional N.5. Enviar mensaje

Para obtener la calidad del funcionamiento de los requisitos funcionales se llevaron a cabo tres iteraciones, las cuales arrojaron los resultados que se muestra en la Tabla 6. Las no conformidades encontradas fueron resueltas en su totalidad.

Iteración	Cantidad de no conformidades	Asociadas a
Primera	7	Interfaz de usuario, manejo de errores
Segunda	3	Interfaz de usuario, manejo de errores
Tercera	0	-

Tabla 6: resultado de las pruebas de validación

3.2.5 Pruebas de sistema

Las pruebas del sistema son una serie de diferentes pruebas cuyo propósito principal es ejercitar por completo el sistema. Aunque cada prueba tenga un propósito diferente, todo el conjunto funciona para verificar que los elementos del sistema se hayan integrado de manera adecuada y que se realicen las funciones asignadas [12]. A continuación, se presentan los tipos de pruebas de sistema que fueron aplicadas a la solución:

- **Prueba de recuperación:** es una prueba del sistema que fuerza al software a fallar en varias formas y que verifica que la recuperación se realice de manera adecuada[12]. Esta prueba se realizó con el objetivo de verificar que los fallos que se puedan presentar en el componente desarrollado no entorpezcan el correcto funcionamiento de la Plataforma “Bienestar”.
- **Prueba de seguridad:** intenta verificar que los mecanismos de protección que se construyenen un sistema en realidad lo protegerán de cualquier penetración impropia[12]. El desarrollo de esta prueba permitió verificar que la información que se transmite por el componente se encuentra debidamente protegida mediante cifrado.

- **Prueba de rendimiento:** se diseña para poner a prueba el rendimiento del software en tiempo de corrida, dentro del contexto de un sistema integrado [12]. Esta prueba permitió comprobar que los tiempos de respuesta del componente están dentro del límite permisible para no entorpecer el correcto funcionamiento de la Plataforma “Bienestar”.

3.3 Validación de la investigación

Para realizar la validación de la investigación se aplicó el método criterio de expertos a travésdel cumplimiento de los pasos siguientes:

1. Selección de los expertos.
2. Realización del cuestionario a los expertos.
3. Aplicación del escalamiento de Likert.

Selección los expertos:para la identificación de los expertos se tuvo en cuentala experiencia profesional en el tema, de modo que tuviera la capacidad de ofrecer valoraciones y hacerrecomendaciones pertinentes en relación con los aspectos que les serían consultados.

Nombre y apellidos	Titulo	Cargo	Años de experiencia
Julio César Vega Iglesias	Master en Informática Aplicada	Jefe de Centro	31
Raciel Garrido Torres	Ingeniero en Ciencias Informáticas	Director de UEB, Empresa XETID	12
Persy Morell Guerra	Ingeniero en Ciencias Informática	Especialista A en Ciencias Informáticas	12
Ydairis Barrera Pérez	Ingeniera Informática	Especialista A en Ciencias Informáticas	7
Niurka Socarrás Hernández	Ingeniero en Ciencias Informáticas	Especialista A en Ciencias Informáticas	5
Pedro Justo Placencia Díaz	Ingeniero en Ciencias Informáticas	Especialista A en Ciencias Informáticas	4
Evelyn Gómez Betancourt	Tec. Medio en Informática	Especialista C enCiencias Informáticas	2

Tabla 7: Expertos seleccionados

Realización del cuestionario a los expertos:una vez seleccionados los expertos se les realizó el cuestionario que se muestra a continuación, compuesto por cinco (5) afirmaciones evaluativas que

permiten conocer la opinión delexperto. Los parámetros evaluativos empleados fueron: MA (muy de acuerdo), DA (de acuerdo), N (Ni deacuerdo ni en desacuerdo), ED (en desacuerdo), CD (completamente en desacuerdo).

No.	Aspecto
1	Las funcionalidades definidas en la propuesta permiten una correcta comunicación de los usuarios dentro de la Plataforma “Bienestar”.
2	La propuesta desarrollada es idóneapara integrar a la Plataforma “Bienestar”.
3	El diseño gráfico de la interfaz es adecuado para los usuarios a la que va dirigida la solución.
4	El componente desarrollado influye positivamente en la interacción entre los actores que intervienen los procesos que se gestionan en la Plataforma “Bienestar”, agilizando la toma de decisiones.
5	La aplicación de la propuesta desarrollada agrega valor a la Plataforma “Bienestar”.

Tabla 8: Cuestionario aplicado

Los parámetros evaluativos se cuantificaron a través de la siguiente escala:

- Muy de acuerdo (100)
- De acuerdo (75)
- Ni de acuerdo ni en desacuerdo (50)
- En desacuerdo (25)
- Completamente en desacuerdo (0)

Aplicación del escalamiento de Likert:para el procesamiento de los resultados se empleó un método que consiste en identificar la media de las diferentes categorías de la escala Likert. Dicho método se aplicó por separado a cada una de losaspectos. Luego se obtuvo la media de las

valoraciones individuales para obtener una valoración integral del sistema. el resultado se muestra en la siguiente tabla:

Experto	Aspectos				
	1	2	3	4	5
Julio César Vega Iglesias	100	100	75	100	100
Raciel Garrido Torres	100	100	100	100	100
Persy Morell Guerra	75	75	75	100	100
Ydairis Barrera Pérez	100	75	75	75	100
Niurka Socarrás Hernández	100	75	75	100	100
Pedro Justo Placencia Díaz	100	75	75	75	100
Evelyn Gómez Betancourt	100	75	75	100	100
Media por aspectos	100	82.1	78.6	92.9	100
Media general	**90.8**				

Tabla 9: Resultado de la aplicación del escalamiento de Likert

La aplicación del escalamiento Likert evidencia que la propuesta solución tiene una alta valoración porparte de los expertos. Se obtuvo un resultadosatisfactorio, con un valor de aprobación superior a un 90 por parte de los expertos.

3.4 Conclusiones

En el presente capítulo se abordaron los elementos de la fase de construcción de la metodología PRODESOFT, a partir de los cuales se pueden arribar a las siguientes conclusiones:

- La identificación de estándares y buenas prácticas de codificación guiaron un desarrollo claro yfluido, cumpliendo con lo establecido en el desarrollo de la Plataforma “Bienestar”.
- Las pruebas de validación identificaron 7 no conformidades en la primera iteración y 3 en la segunda, para una tercera iteración se demostró que todas fueron resueltas.
- Las pruebas de sistema permitieron verificar que el componente desarrollado no impacta negativamente en el funcionamiento y rendimiento de la Plataforma “Bienestar”.
- La validación de la investigación aplicando el método Criterio de Expertos arrojó resultados favorablescon un valor superior a un 90 de aprobación.

Conclusiones generales

Una vez culminado el presente trabajo de diploma se puede concluir que se le dio cumplimiento al objetivo general de la investigación, resaltando que:

- La elaboración del marco teórico conceptual de la investigación permitió sentar las bases de la misma.
- Se realizó un estudio de sistemas de comunicación sincrónica en Cuba y el mundo, aportando ideas al componente que posteriormente fue desarrollado.
- La modelación y especificación de requisitos de software de la propuesta de solución permitió identificar 10 requisitos funcionales y 6 no funcionales que posteriormente fueron implementados.
- El uso de patrones GRASP y los cuatro patrones de diseño GOF identificados se cohesionan al estilo arquitectónico modelo-vista-controlador agregando a la solución un alto grado de flexibilidad ante posibles modificaciones.
- Las pruebas permitieron detectar y corregir los errores no detectados durante la implementación, verificando además que el componente desarrollado no influye negativamente en el correcto funcionamiento de la Plataforma “Bienestar”.
- La validación de la investigación aplicando el método Criterio de Expertos arrojó resultados favorables con un valor superior a un 90de aprobación.

Referencias bibliográficas

[1] O. Pérez Salomón, «Gobierno Electrónico: Un mecanismo de participación popular», *Cubadebate*, 12-jun-2018. [En línea]. Disponible en: http://www.cubadebate.cu/opinion/2018/06/12/gobierno-electronico-un-mecanismo-de-participacion-popular/. [Accedido: 28-ene-2020].

[2] «La comunicacion». [En línea]. Disponible en: https://www.slideshare.net/NTICProfesor/la-comunicacion-7973722. [Accedido: 29-ene-2020].

[3] J. Andreu, *Voz IP (Servicios en red)*. Editex, 2011.

[4] «Home | ProcessMaker». [En línea]. Disponible en: https://www.processmaker.com/. [Accedido: 30-ene-2020].

[5] «Características | Averigua lo que Skype puede hacer por ti | Skype». [En línea]. Disponible en: https://www.skype.com/es/features/. [Accedido: 29-ene-2020].

[6] Unknown, «New Google+: Stream, Hangouts, and Photos», *New Google+*, 15-may-2013. [En línea]. Disponible en: http://googleplusproject.blogspot.com/2013/05/new-google-stream-hangouts-and-photos.html. [Accedido: 29-ene-2020].

[7] «Google Hangouts se cierra ante sus competidores y no usa XMPP», *El Español*, 16-may-2013. [En línea]. Disponible en: https://www.elespanol.com/omicrono/20130516/google-hangouts-cierra-competidores-no-usa-xmpp/9249129_0.html. [Accedido: 29-ene-2020].

[8] «FAQ de WhatsApp», *WhatsApp.com*. [En línea]. Disponible en: https://faq.whatsapp.com/es/general. [Accedido: 29-ene-2020].

[9] «WhatsApp alcanza los 2 000 millones de usuarios», *Cubadebate*, 12-feb-2020. [En línea]. Disponible en: http://www.cubadebate.cu/noticias/2020/02/12/whatsapp-alcanza-los-2-000-millones-de-usuarios/. [Accedido: 13-feb-2020].

[10] XETID, «Ficha técnica de Platel». .

[11] G. Camarillo, *SIP Demystified*. McGraw Hill Professional, 2001.

[12] R. S. Pressman y B. R. Maxim, *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education, 2015.

[13] XETID, «Proceso del desarrollo del software PRODESOFT». .

[14] «What is Unified Modeling Language (UML)? - Definition from Techopedia», *Techopedia.com*. [En línea]. Disponible en: https://www.techopedia.com/definition/3243/unified-modeling-language-uml. [Accedido: 28-ene-2020].

[15] «Visual Paradigm Product Overview». [En línea]. Disponible en: https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html. [Accedido: 29-ene-2020].

[16] «Apache Subversion». [En línea]. Disponible en: http://subversion.apache.org/. [Accedido: 29-ene-2020].

[17] G. Rooney y D. Berlin, *Practical Subversion*. Apress, 2007.

[18] «¿Qué es un lenguaje de programación y qué tipos existen?», *Rock Content*, 20-abr-2019. [En línea]. Disponible en: https://rockcontent.com/es/blog/que-es-un-lenguaje-de-programacion/. [Accedido: 29-ene-2020].

[19] «PHP: Hypertext Preprocessor». [En línea]. Disponible en: https://www.php.net/. [Accedido: 29-ene-2020].

[20] «JavaScript», *Documentación web de MDN*. [En línea]. Disponible en: https://developer.mozilla.org/es/docs/Web/JavaScript. [Accedido: 19-jul-2020].

[21] «Definición de JavaScript», *Definición ABC*. [En línea]. Disponible en: https://www.definicionabc.com/tecnologia/javascript.php. [Accedido: 29-ene-2020].

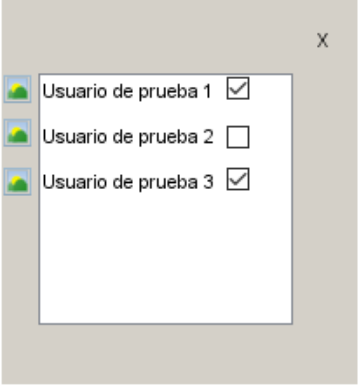
[22] «Definición de HTML», *Definición ABC*. [En línea]. Disponible en: https://www.definicionabc.com/tecnologia/html.php. [Accedido: 29-ene-2020].
[23] «HTML & CSS - W3C». [En línea]. Disponible en: https://www.w3.org/standards/webdesign/htmlcss. [Accedido: 19-jul-2020].
[24] «Capítulo 1. Introducción (Introducción a CSS)». [En línea]. Disponible en: https://uniwebsidad.com/libros/css/capitulo-1?from=librosweb. [Accedido: 29-ene-2020].
[25] «Biblioteca informática - LOGICA DE PROGRAMACION». [En línea]. Disponible en: https://sites.google.com/site/israelcortess/documentos/biblioteca-informatica-3. [Accedido: 29-ene-2020].
[26] «Vue.js». [En línea]. Disponible en: https://vuejs.org/. [Accedido: 29-ene-2020].
[27] «Gulliver: Programmer's Reference Guide | Documentation@ProcessMaker». [En línea]. Disponible en: https://wiki.processmaker.com/index.php/Gulliver%3A_Programmer%2527s_Reference_Guide. [Accedido: 26-feb-2020].
[28] «JSON». [En línea]. Disponible en: https://www.json.org/json-es.html. [Accedido: 29-ene-2020].
[29] «PhpStorm: el IDE rápido e inteligente para programación en PHP de JetBrains», *JetBrains*. [En línea]. Disponible en: https://www.jetbrains.com/phpstorm/. [Accedido: 29-ene-2020].
[30] Larman, Craig, *UML y patrones. Introducción al análisis y diseño orientado a objetos*. Prentice Hall, 1999.
[31] I. Sommerville, *Software engineering*. Boston: Pearson, 2016.
[32] «ProcessMaker Architecture Diagrams | Documentation@ProcessMaker». [En línea]. Disponible en: https://wiki.processmaker.com/index.php/ProcessMaker_Architecture_Diagrams. [Accedido: 25-feb-2020].
[33] L. Bass, P. Clements, y R. Kazman, *Software Architecture in Practice: Software Architect Practice_c3*. Addison-Wesley, 2012.
[34] R. Johnson, E. Gamma, J. Vlissides, y R. Helm, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
[35] C. A. Guerrero, J. M. Suárez, y L. E. Gutiérrez, «Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web», *Inf. Tecnológica*, vol. 24, n.º 3, pp. 103-114, 2013, doi: 10.4067/S0718-07642013000300012.
[36] A. Osmani, *Learning JavaScript Design Patterns: A JavaScript and jQuery Developer's Guide*. O'Reilly Media, Inc., 2012.

Anexo 1: Descripción de requisitos funcionales

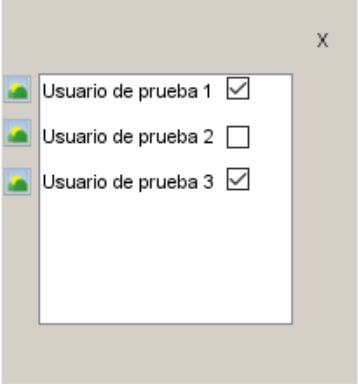
RF.01 Registrar usuario en la Plataforma Platel: permite el registro automático del usuario en la Plataforma “Platel”.

Conceptos tratados	**Conceptos**	**Atributos**
	Usuario	usuario, contraseña
Precondiciones	**Precondiciones**	**Pre-requisitos**
	El usuario debe estar autenticado en la Plataforma “Bienestar”	-
	El usuario debe tener acceso a la funcionalidad.	
Descripción	1- El usuario se autentica correctamente en la Plataforma “Bienestar”. 2- El sistema registra al usuario en la Plataforma Platel.	
Complejidad	Crítico	
Validaciones	Ninguna	
Post-condiciones	Se ha registrado el usuario en la Plataforma Platel.	
Post-requisitos	Ninguno	
Prototipo de interfaz de usuario	-	

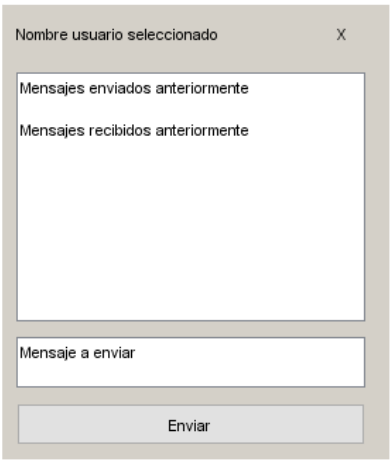
RF.02 Mostrar lista de usuarios:el componente debe permitir mostrar una lista con el nombre completo de todos los usuarios de la Plataforma “Bienestar”.

<table>
<tr><td rowspan="2">Conceptos tratados</td><td>Conceptos</td><td>Atributos</td></tr>
<tr><td>Usuario</td><td>usuario, nombre, apellidos, foto</td></tr>
<tr><td rowspan="3">Precondiciones</td><td>Precondiciones</td><td>Pre-requisitos</td></tr>
<tr><td>El usuario debe estar autenticado</td><td rowspan="2">Registrar usuario en la Plataforma Platel.</td></tr>
<tr><td>El usuario debe tener acceso a la funcionalidad.</td></tr>
<tr><td>Descripción</td><td colspan="2">1- El usuario se autentica correctamente en la Plataforma “Bienestar”.
2- El sistema muestra una ventana en la esquina inferior derecha con la lista de usuarios, sus fotos y si está conectado o no en ese momento.</td></tr>
<tr><td>Complejidad</td><td colspan="2">Crítico</td></tr>
<tr><td>Validaciones</td><td colspan="2">Ninguna</td></tr>
<tr><td>Post-condiciones</td><td colspan="2">Se muestra la lista de usuarios y sus estados.</td></tr>
<tr><td>Post-requisitos</td><td colspan="2">Ninguno</td></tr>
<tr><td>Prototipo de interfaz de usuario</td><td colspan="2"> 
</td></tr>
</table>

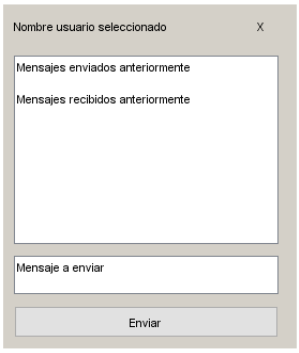
RF.03 Mostrar presencia de los usuarios:el componente debe mostrar si un usuario está conectado o no en el momento.

<table>
<tr><td rowspan="2">Conceptos tratados</td><td>Conceptos</td><td>Atributos</td></tr>
<tr><td>Usuario</td><td>usuario, estado</td></tr>
<tr><td rowspan="3">Precondiciones</td><td>Precondiciones</td><td>Pre-requisitos</td></tr>
<tr><td>El usuario debe estar autenticado</td><td rowspan="2">Registrar usuario en la Plataforma Platel.</td></tr>
<tr><td>El usuario debe tener acceso a la funcionalidad.</td></tr>
<tr><td>Descripción</td><td colspan="2">1- El usuario se autentica correctamente en la Plataforma “Bienestar”.
2- El sistema la lista de usuarios y si está conectado o no en ese momento.</td></tr>
<tr><td>Complejidad</td><td colspan="2">Crítico</td></tr>
<tr><td>Validaciones</td><td colspan="2">Ninguna</td></tr>
<tr><td>Post-condiciones</td><td colspan="2">Se muestra el estado de los usuarios.</td></tr>
<tr><td>Post-requisitos</td><td colspan="2">Ninguno</td></tr>
<tr><td>Prototipo de interfaz de usuario</td><td colspan="2"> 
</td></tr>
</table>

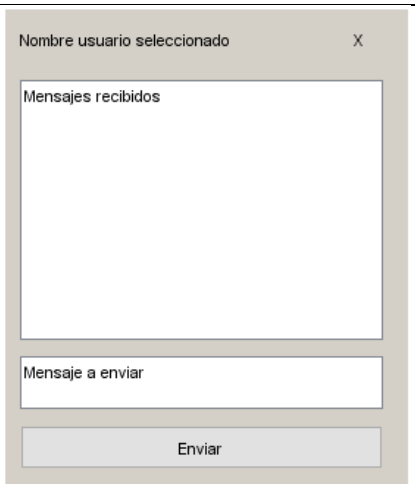
RF.04 Mostrar conversaciones: a partir de la selección de un usuario, el componente debe mostrar una ventana con las conversaciones que se hayan realizado con el mismo.

<table>
<tr><td rowspan="2">Conceptos tratados</td><td>Conceptos</td><td>Atributos</td></tr>
<tr><td>Mensaje</td><td>origen, contenido</td></tr>
<tr><td rowspan="3">Precondiciones</td><td>Precondiciones</td><td>Pre-requisitos</td></tr>
<tr><td>El usuario debe estar autenticado</td><td rowspan="2">Registrar usuario en la Plataforma Platel.
Mostrar lista de usuarios.</td></tr>
<tr><td>El usuario debe tener acceso a la funcionalidad.</td></tr>
<tr><td>Descripción</td><td colspan="2">1- El usuario se autentica correctamente en la Plataforma “Bienestar”.
2- El sistema muestra una ventana en la esquina inferior derecha con la lista de usuarios, sus fotos y si está conectado o no en ese momento.
3- El usuario selecciona a un usuario de la lista de usuarios mostrada.
4- El sistema muestra la interfaz mensajes con el usuario seleccionado, donde se muestran las conversaciones realizadas anteriormente.</td></tr>
<tr><td>Complejidad</td><td colspan="2">Crítico</td></tr>
<tr><td>Validaciones</td><td colspan="2">Ninguna</td></tr>
<tr><td>Post-condiciones</td><td colspan="2">Se muestran los mensajes anteriores.</td></tr>
<tr><td>Post-requisitos</td><td colspan="2">Ninguno</td></tr>
<tr><td>Prototipo de interfaz de usuario</td><td colspan="2"> 
</td></tr>
</table>

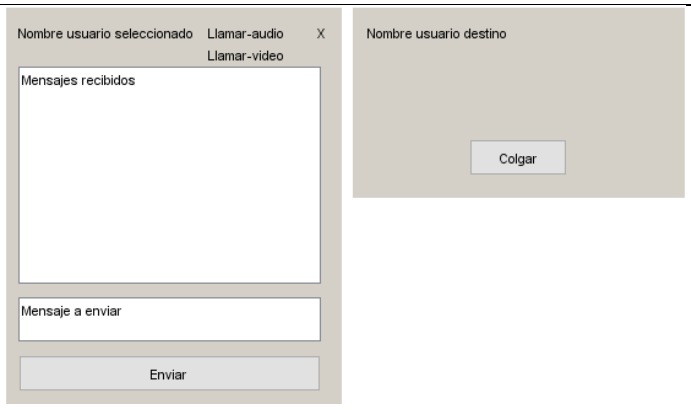
RF.05 Enviar mensaje:el componente debe permitir a un usuario enviar mensajes de texto en tiempo real a otro usuario conectado.

<table>
<tr><td>Conceptos tratados</td><td>Conceptos</td><td>Atributos</td></tr>
<tr><td></td><td>Mensaje</td><td>destinatario, contenido</td></tr>
<tr><td>Precondiciones</td><td>Precondiciones</td><td>Pre-requisitos</td></tr>
<tr><td></td><td>El usuario debe estar autenticado.</td><td rowspan="2">Registrar usuario en la Plataforma Platel.
Mostrar lista de usuarios.
Mostrar conversaciones.</td></tr>
<tr><td></td><td>El usuario debe tener acceso a la funcionalidad.</td></tr>
<tr><td>Descripción</td><td colspan="2">9- El usuario se autentica correctamente en la Plataforma “Bienestar”.
10- El sistema muestra una ventana en la esquina inferior derecha con la lista de usuarios, sus fotos y si está conectado o no en ese momento.
11- El usuario selecciona a un usuario de la lista de usuarios disponible.
12- El sistema muestra la interfaz mostrar conversaciones con el usuario seleccionado.
13- El usuario escribe el texto a enviar en el campo correspondiente.
14- El usuario presiona el botón Enviar o presiona la tecla Enter.
15- El sistema envía el mensaje al usuario correspondiente.
16- El sistema muestra el nuevo mensaje enviado al final de la lista de mensajes.</td></tr>
<tr><td>Complejidad</td><td colspan="2">Crítico</td></tr>
<tr><td>Validaciones</td><td colspan="2">Ninguna</td></tr>
<tr><td>Post-condiciones</td><td colspan="2">Se ha enviado el mensaje.</td></tr>
<tr><td>Post-requisitos</td><td colspan="2">Ninguno</td></tr>
<tr><td>Prototipo de interfaz de usuario</td><td colspan="2"> 
</td></tr>
</table>

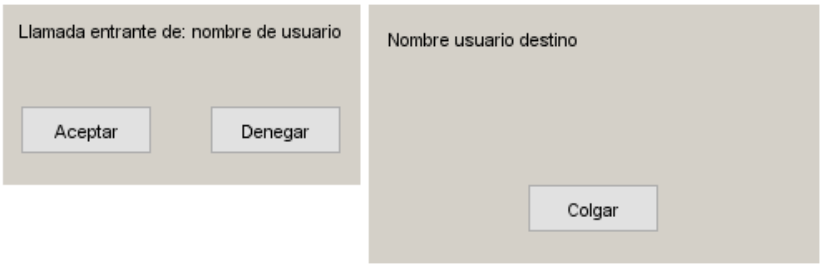
RF.06 Recibir mensaje:el componente debe permitir recibir mensajes de texto en tiempo real.

<table>
<tr><td>Conceptos tratados</td><td>Conceptos</td><td>Atributos</td></tr>
<tr><td></td><td>Mensaje</td><td>origen, contenido</td></tr>
<tr><td>Precondiciones</td><td>Precondiciones</td><td>Pre-requisitos</td></tr>
<tr><td></td><td>El usuario debe estar autenticado.</td><td rowspan="2">Registrar usuario en la Plataforma Platel.
Mostrar lista de usuarios.
Mostrar conversaciones.</td></tr>
<tr><td></td><td>El usuario debe tener acceso a la funcionalidad.</td></tr>
<tr><td>Descripción</td><td colspan="2">1- El usuario se autentica correctamente en la Plataforma “Bienestar”.
2- El sistema muestra una ventana en la esquina inferior derecha con la lista de usuarios, sus fotos y si está conectado o no en ese momento.
3- Al recibir un mensaje el sistema muestra el usuario que lo envía de forma parpadeante en la lista de usuarios.
4- El usuario selecciona al usuario parpadeante de la lista de usuarios disponible.
5- El sistema muestra la interfaz mostrar conversaciones con el usuario seleccionado, mostrando así el mensaje recibido.</td></tr>
<tr><td>Complejidad</td><td colspan="2">Crítico</td></tr>
<tr><td>Validaciones</td><td colspan="2">Ninguna</td></tr>
<tr><td>Post-condiciones</td><td colspan="2">Se ha recibido el mensaje.</td></tr>
<tr><td>Post-requisitos</td><td colspan="2">Ninguno</td></tr>
<tr><td>Prototipo de interfaz de usuario</td><td colspan="2"> 
</td></tr>
</table>

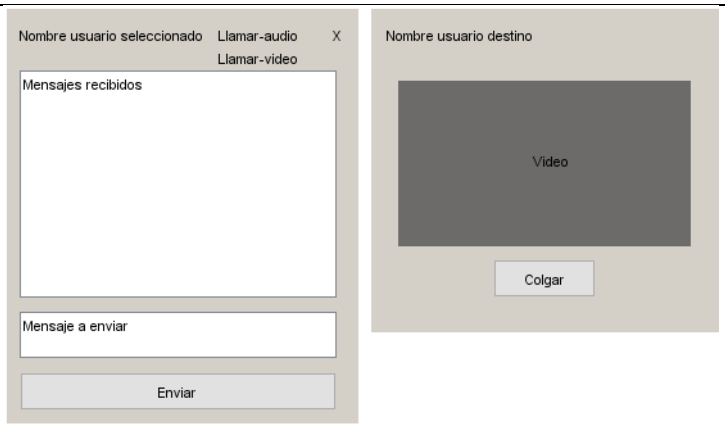
RF.07 Realizar llamada de audio: el componente debe permitir realizar llamadas de audio en tiempo real.

<table>
<tr><td>Conceptos tratados</td><td>Conceptos</td><td>Atributos</td></tr>
<tr><td></td><td>Llamada</td><td>destinatario</td></tr>
<tr><td>Precondiciones</td><td>Precondiciones</td><td>Pre-requisitos</td></tr>
<tr><td></td><td>El usuario debe estar autenticado.</td><td rowspan="2">Registrar usuario en la Plataforma Platel.
Mostrar lista de usuarios.
Mostrar conversaciones.</td></tr>
<tr><td></td><td>El usuario debe tener acceso a la funcionalidad.</td></tr>
<tr><td>Descripción</td><td colspan="2">1- El usuario se autentica correctamente en la Plataforma “Bienestar”.
2- El sistema muestra una ventana en la esquina inferior derecha con la lista de usuarios, sus fotos y si está conectado o no en ese momento.
3- El usuario selecciona a un usuario de la lista de usuarios disponible.
4- El sistema muestra la interfaz mostrar conversaciones con el usuario seleccionado.
5- El usuario selecciona la opción llamada de audio.
6- El sistema muestra la interfaz de llamadas.</td></tr>
<tr><td>Complejidad</td><td colspan="2">Crítico</td></tr>
<tr><td>Validaciones</td><td colspan="2">Ninguna</td></tr>
<tr><td>Post-condiciones</td><td colspan="2">Se ha realizado la llamada de audio.</td></tr>
<tr><td>Post-requisitos</td><td colspan="2">Ninguno</td></tr>
<tr><td>Prototipo de interfaz de usuario</td><td colspan="2"> 
</td></tr>
</table>

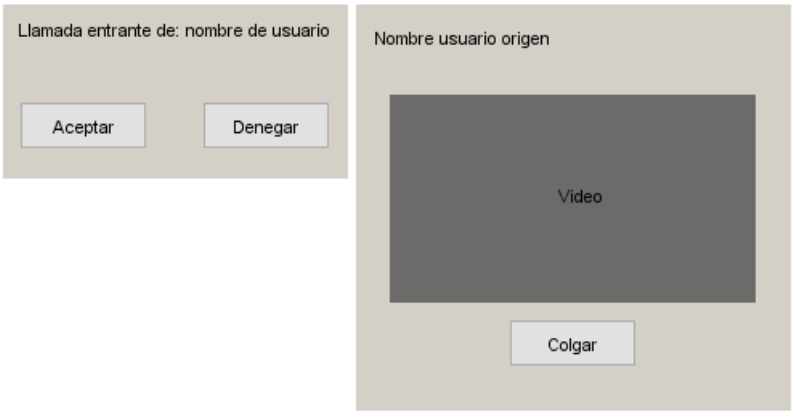
RF.08 Recibir llamada de audio: el componente debe permitir recibir llamadas de audio en tiempo real.

<table>
<tr><td rowspan="2">Conceptos tratados</td><td>Conceptos</td><td>Atributos</td></tr>
<tr><td>Llamada</td><td>origen</td></tr>
<tr><td rowspan="3">Precondiciones</td><td>Precondiciones</td><td>Pre-requisitos</td></tr>
<tr><td>El usuario debe estar autenticado.</td><td rowspan="2">Registrar usuario en la Plataforma Platel.
Mostrar lista de usuarios.</td></tr>
<tr><td>El usuario debe tener acceso a la funcionalidad.</td></tr>
<tr><td>Descripción</td><td colspan="2">1- El usuario se autentica correctamente en la Plataforma “Bienestar”.
2- Al recibir una llamada el sistema muestra al usuario una interfaz con la opción de aceptar o denegar.
3- El usuario selecciona la opción aceptar llamada.
4- El sistema muestra la interfaz de llamadas.</td></tr>
<tr><td>Complejidad</td><td colspan="2">Crítico</td></tr>
<tr><td>Validaciones</td><td colspan="2">Ninguna</td></tr>
<tr><td>Post-condiciones</td><td colspan="2">Se ha recibidola llamada de audio.</td></tr>
<tr><td>Post-requisitos</td><td colspan="2">Ninguno</td></tr>
<tr><td>Prototipo de interfaz de usuario</td><td colspan="2"> 
</td></tr>
</table>

RF.09 Realizar videollamada: el componente debe permitir realizar videollamadas en tiempo real.

<table>
<tr><td>Conceptos tratados</td><td>Conceptos</td><td>Atributos</td></tr>
<tr><td></td><td>Llamada</td><td>destinatario</td></tr>
<tr><td>Precondiciones</td><td>Precondiciones</td><td>Pre-requisitos</td></tr>
<tr><td></td><td>El usuario debe estar autenticado.</td><td rowspan="2">Registrar usuario en la Plataforma Platel.
Mostrar lista de usuarios.
Mostrar conversaciones.</td></tr>
<tr><td></td><td>El usuario debe tener acceso a la funcionalidad.</td></tr>
<tr><td>Descripción</td><td colspan="2">1- El usuario se autentica correctamente en la Plataforma “Bienestar”.
2- El sistema muestra una ventana en la esquina inferior derecha con la lista de usuarios, sus fotos y si está conectado o no en ese momento.
3- El usuario selecciona a un usuario de la lista de usuarios disponible.
4- El sistema muestra la interfaz mostrar conversaciones con el usuario seleccionado.
5- El usuario selecciona la opción llamada de audio.
6- El sistema muestra la interfaz de llamadas.</td></tr>
<tr><td>Complejidad</td><td colspan="2">Crítico</td></tr>
<tr><td>Validaciones</td><td colspan="2">Ninguna</td></tr>
<tr><td>Post-condiciones</td><td colspan="2">Se ha realizado la llamada de video.</td></tr>
<tr><td>Post-requisitos</td><td colspan="2">Ninguno</td></tr>
<tr><td>Prototipo de interfaz de usuario</td><td colspan="2"> 
</td></tr>
</table>

RF.10 Recibir videollamada: el componente debe permitir recibir videollamadas en tiempo real.

Conceptos tratados	**Conceptos**	**Atributos**
	Llamada	origen
Precondiciones	**Precondiciones**	**Pre-requisitos**
	El usuario debe estar autenticado.	Registrar usuario en la Plataforma Platel. Mostrar lista de usuarios.
	El usuario debe tener acceso a la funcionalidad.	
Descripción	1- El usuario se autentica correctamente en la Plataforma “Bienestar”. 2- Al recibir una llamada el sistema muestra al usuario una interfaz con la opción de aceptar o denegar. 3- El usuario selecciona la opción aceptar llamada. 4- El sistema muestra la interfaz de llamadas.	
Complejidad	Crítico	
Validaciones	Ninguna	
Post-condiciones	Se ha recibido la llamada de video.	
Post-requisitos	Ninguno	
Prototipo de interfaz de usuario		

Anexo 2: Casos de pruebas de validación

RF.01 Registrar usuario en la Plataforma Platel

Condiciones de ejecución	Ninguna.
Descripción de la prueba	Comprobar que el sistema registre al usuario correctamente.
Pasos de ejecución	1. El usuario probador1 se autentica a la Plataforma “Bienestar”. 2. El operador verifica que el componente muestre al usuario probador1 como conectado.
Resultados esperados	El usuario probador1 es registrado de forma correcta en la Plataforma “Platel”.
Evaluación	Prueba satisfactoria

Tabla 10: Caso de prueba 1 para el RF.1

RF.02 Mostrar lista de usuarios

Condiciones de ejecución	El usuario está autenticado al sistema y tiene acceso a la funcionalidad.
Descripción de la prueba	Comprobar que el sistema muestre la lista de usuarios de forma correcta.
Pasos de ejecución	1. Los usuarios probador1 y probador2 se autentican en el sistema desde dos dispositivos diferentes. 2. El operador verifica que el componente muestre la lista de usuarios con todos los usuarios existentes y sus fotos.
Resultados esperados	El sistema muestra la lista de usuarios de forma correcta.
Evaluación	Prueba satisfactoria

Tabla 11: Caso de prueba 1 para el RF.2

RF.03 Mostrar presencia de los usuarios

Condiciones de ejecución	El usuario está autenticado al sistema y tiene acceso a la funcionalidad.
Descripción de la prueba	Comprobar que el sistema muestre la presencia de los usuarios de forma correcta.
Pasos de ejecución	1. Los usuarios probador1 y probador2 se autentican en el sistema desde dos dispositivos diferentes. 2. El operador desde el usuario probador1 verifica que el estado de todos los usuarios sea desconectado menos el de probador2 que debe aparecer conectado. 3. El operador desde el usuario probador2 verifica que el estado de todos los usuarios sea desconectado menos el de probador1 que debe aparecer conectado.
Resultados esperados	El sistema muestra la lista de usuarios de forma correcta.
Evaluación	Prueba satisfactoria

Tabla 12: Caso de prueba 1 para el RF.3

RF.04 Mostrar conversaciones

Condiciones de ejecución	El usuario está autenticado al sistema y tiene acceso a la funcionalidad.
Descripción de la prueba	Comprobar que el sistema muestre las conversaciones anteriores de forma correcta.
Pasos de ejecución	1. El usuario probador1 se autentica al sistema. 2. El usuario probador1 selecciona al usuario probador2 en la lista de usuarios. 3. El operador verifica que las conversaciones anteriores se muestren de forma correcta.
Resultados esperados	El sistema muestra las conversaciones anteriores.
Evaluación	Prueba satisfactoria

Tabla 13: Caso de prueba 1 para el RF. 4

RF.05 Enviar mensajey

RF.06 Recibir mensaje

Condiciones de ejecución	El usuario está autenticado al sistema y tiene acceso a la funcionalidad.
Descripción de la prueba	Comprobar que el sistema envié el mensaje correctamente.
Pasos de ejecución	1. Los usuarios probador1 y probador2 se autentican en el sistema desde dos dispositivos diferentes. 2. El usuario probador1 selecciona al usuario probador2 en la lista de usuarios. 3. El usuario probador1 envía un mensaje al usuario probador2. 4. El usuario probador2 comprueba que el usuario probador1 aparezca parpadeando en la lista de usuarios. 5. El usuario probador2 selecciona al usuario probador1 en la lista de usuarios y verifica que el mensaje enviado por probador1 aparezca correctamente.
Resultados esperados	El usuario probador2 recibe el mensaje correctamente.
Evaluación	Prueba satisfactoria

Tabla 14: Caso de prueba 1 para los RF 5 y 6

RF.07 Realizar llamada de audioy

RF.08 Recibir llamada de audio

Condiciones de ejecución	El usuario está autenticado al sistema y tiene acceso a la funcionalidad.
Descripción de la prueba	Comprobar que el sistema realice llamadas de audio de forma correcta.
Pasos de ejecución	1. Los usuarios probador1 y probador2 se autentican en el sistema desde dos dispositivos diferentes. 2. El usuario probador1 selecciona al usuario probador2 en la lista de usuarios. 3. El usuario probador1 selecciona la opción de llamada de audio. 4. El usuario probador2 comprueba le salga la pantalla de llamada entrante. 5. El usuario probador2 selecciona la opción de recibir llamada. 6. Se verifica que el sistema muestre la ventana de llamada y se realice la llamada de forma correcta enviando y recibiendo el audio captado por el micrófono de cada dispositivo.
Resultados esperados	La llamada de audio se realiza de forma correcta.
Evaluación	Prueba satisfactoria

Tabla 15: Caso de prueba 1 para los RF 7 y 8

Condiciones de ejecución	El usuario está autenticado al sistema y tiene acceso a la funcionalidad.
Descripción de la prueba	Comprobar que el sistema realice llamadas de audio de forma correcta.
Pasos de ejecución	1. Los usuarios probador1 y probador2 se autentican en el sistema desde dos dispositivos diferentes. 2. El usuario probador1 selecciona al usuario probador2 en la lista de usuarios. 3. El usuario probador1 selecciona la opción de llamada de audio. 4. El usuario probador2 comprueba le salga la pantalla de llamada entrante. 5. El usuario probador2 selecciona la opción de denegar llamada. 6. Se verifica que el sistema notifique al usuario probador1 de que el usuario probador2 ha denegado la llamada.
Resultados esperados	La llamada de audio es denegada y el usuario probador1 es notificado.
Evaluación	Prueba satisfactoria

Tabla 16: Caso de prueba 2 para los RF 7 y 8

RF.09 Realizar videollamaday

RF.10 Recibir videollamada

Condiciones de ejecución	El usuario está autenticado al sistema y tiene acceso a la funcionalidad.
Descripción de la prueba	Comprobar que el sistema realice videollamadas de forma correcta.
Pasos de ejecución	1. Los usuarios probador1 y probador2 se autentican en el sistema desde dos dispositivos diferentes. 2. El usuario probador1 selecciona al usuario probador2 en la lista de usuarios. 3. El usuario probador1 selecciona la opción de videollamada. 4. El usuario probador2 comprueba le salga la pantalla de llamada entrante. 5. El usuario probador2 selecciona la opción de recibir llamada. 6. Se verifica que el sistema muestre la ventana de llamada y se realice la llamada de forma correcta enviando y recibiendo el audio y video captado por el micrófono y cámara de cada dispositivo.
Resultados esperados	La videollamada se realiza de forma correcta.
Evaluación	Prueba satisfactoria

Tabla 17: Caso de prueba 1 para los RF 9 y 10

Condiciones de ejecución	El usuario está autenticado al sistema y tiene acceso a la funcionalidad.
Descripción de la prueba	Comprobar que el sistema realice videollamadas de forma correcta.
Pasos de ejecución	1. Los usuarios probador1 y probador2 se autentican en el sistema desde dos dispositivos diferentes. 2. El usuario probador1 selecciona al usuario probador2 en la lista de usuarios. 3. El usuario probador1 selecciona la opción de videollamada. 4. El usuario probador2 comprueba le salga la pantalla de llamada entrante. 5. El usuario probador2 selecciona la opción de denegar llamada. 6. Se verifica que el sistema notifique al usuario probador1 de que el usuario probador2 ha denegado la llamada.
Resultados esperados	La llamada de audio es denegada y el usuario probador1 es notificado.
Evaluación	Prueba satisfactoria

Tabla 18: Caso de prueba 2 para los RF 9 y 10

Anexo 3: Encuesta presentada al grupo de expertos para validar el componente propuesto.

Estimado(a) compañero(a):

La presente encuesta forma parte de las acciones para validar el componente de comunicación sincrónica para la Plataforma “Bienestar”. Su análisis y colaboración en cuanto a los aspectos que sometemos a su consideración serán de invaluable ayuda para la validación de la investigación. Le agradecemos de antemano por su valiosa contribución.

Datos generales del encuestado:

Nombre(s) y apellidos:_____

Título universitario: _____

Cargo ocupacional: _____

Años de experiencia: ____

Solicitamos su valoración sobre diferentes aspectos que a continuación se presentan. Para expresar su evaluación, por favor, luego de utilizar el componente, evalúe cada uno de los aspectos que se le presentan en la tabla, marcando con una cruz (X) en la casilla correspondiente.

Tenga en cuenta para ello el siguiente código de categorías de clasificación.

- **MA** (muy de acuerdo)
- **DA** (de acuerdo)
- **N** (Ni de acuerdo ni en desacuerdo)
- **ED** (en desacuerdo)
- **CD** (completamente en desacuerdo)

No.	Aspecto	Valoración				
		MA	DA	N	ED	CD
1	Las funcionalidades definidas en la propuesta permiten una correcta comunicación de los usuarios dentro de la Plataforma “Bienestar”.					
2	La propuesta desarrollada es idóneapara integrar a la Plataforma “Bienestar”.					
3	El diseño gráfico de la interfaz es adecuado para los usuarios a la que va dirigida la solución.					
4	El componente desarrollado influye positivamente en la interacción entre los actores que intervienen los procesos que se gestionan en la Plataforma “Bienestar”, agilizando la toma de decisiones.					
5	La aplicación de la propuesta desarrollada agrega valor a la Plataforma “Bienestar”.					

Si desea exponer cualquier otra opinión, por favor, expréselo en el espacio disponible a continuación:

_____

_____

_____

_____

_____

_____

_____