Pontificia Universidad JAVERIANA Cali

Facultad de Ingeniería y Ciencias

Acta de Correcciones al Proyecto de Grado
Ingeniería de sistemas y computación

Fecha: 20 de julio de 2023

Autores: Julian Paredes Conde

Nombre del Proyecto de Grado: Chatbot para mejorar la comunicación entre tiendas de barrio y sus proveedores.

Director: Juan Pablo García Cifuentes

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.

Firma de Director(a) del Proyecto de Grado

Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Ingeniero de Sistemas y Computación.

DR. HERNÁN CAMILO ROCHA NIÑO
Decano de la Facultad de Ingeniería

ING. GERARDO MAURICIO SARRIA
Director Carrera Ingeniería Sistemas y Computación.

ING. JUAN PABLO GARCÍA CIFUENTES
Director(a) Trabajo

ING. GERARDO MAURICIO SARRIA
Jurado 1

ING. DIEGO LINARES
Jurado 2

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería.
Ingeniería de Sistemas y Computación.
Proyecto de Grado.

Chatbot para mejorar la comunicación entre tiendas de barrio y sus proveedores.

Julian Paredes Conde

Director:
Juan Pablo Garcia Cifuentes

15 de Junio del 2023

Santiago de Cali, 15 de Junio del 2023.

Señores
Pontificia Universidad Javeriana Cali.
Dr. Gerardo Mauricio Sarria Montemiranda
Director Carrera de Ingeniería de Sistemas y Computación.
Cali.

Cordial Saludo.

Por medio de la presente me permito informarle que el estudiante de Ingeniería de Sistemas y Computación Julian Paredes Conde (cod: 8918744) trabaja bajo mi dirección en el proyecto de grado titulado "Chatbot para mejorar la comunicación entre tiendas de barrio y sus proveedores.".

Atentamente,

Juan Pablo Garcia Cifuentes

Santiago de Cali, 15 de Junio del 2023.

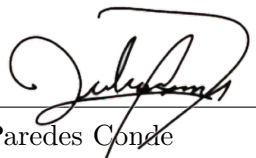
Señores
Pontificia Universidad Javeriana Cali.
Dr. Gerardo Mauricio Sarria Montemiranda
Director Carrera de Ingeniería de Sistemas y Computación.
Cali.

Cordial Saludo.

Me permito presentar a su consideración el proyecto de grado titulado "Chatbot para mejorar la comunicación entre tiendas de barrio y sus proveedores." con el fin de cumplir con los requisitos exigidos por la Universidad para llevar a cabo el proyecto de grado y posteriormente optar al título de Ingeniero de Sistemas y Computación.

Al firmar aquí, doy fe que entiendo y conozco las directrices para la presentación de trabajos de grado de la Facultad de Ingeniería aprobadas el 26 de Noviembre de 2009, donde se establecen los plazos y normas para el desarrollo del anteproyecto y del trabajo de grado.

Atentamente,



Julian Paredes Conde
Código: 8918744

Agradecimientos

Agradezco primeramente a mis padres por ser un apoyo incondicional durante toda mi carrera y por creer en mí siempre, al profesor Juan Pablo García, por atender mis dudas de la mejor manera, por su apoyo, guía e impulso hacia la innovación del uso de nuevas tecnologías, su orientación y supervisión, hacía la ejecución teórica y práctica de los objetivos planteados. A la empresa Distri Romel SA por confiar en mí y en el desarrollo de este proyecto. También quiero agradecer a todos los profesores de la Pontificia Universidad Javeriana Cali que me formaron como profesional e hicieron de mí una mejor persona.

Resumen

La tecnología es una herramienta que resulta ser de suma importancia hoy en día, la cual nos permite comunicarnos con las demás personas y darnos a conocer para así generar un mayor impacto en un público objetivo, al trabajar en conjunto con las estrategias de publicidad o marketing existentes. Asimismo, la tecnología puede llegar a ser fundamental para impulsar o controlar un negocio, ya que esta consigue aumentar su productividad y eficiencia, al lograr romper la barrera que existe entre un proveedor y su cliente para comunicarse a la distancia. En Colombia se han venido implementando estrategias digitales en las empresas muy lentamente en comparación con otros países [1]. De hecho, el COVID-19 y el paro nacional fueron detonantes que obligaron a los negocios, como lo son las tiendas de barrio y sus proveedores, a adaptarse a una nueva era digital en la cual estos pueden llegar a vender, comprar o distribuir sus productos de una manera más eficiente. Sin embargo, dada la crisis económica, la falta de recursos, de comunicación y el desabastecimiento que se estaba viviendo en dicho momento, ha sido difícil generar una transición digital en muchas tiendas de barrio y proveedores de las mismas que lo requieren para impulsar su negocio. Dado que, aunque se han desarrollado distintas aplicaciones móviles o web que buscan servir como intermediarias, estas no abarcan las necesidades de la mayoría de personas o negocios.

El presente trabajo de grado buscará crear una solución tecnológica por medio de un Chatbot que permita mejorar la comunicación entre los tenderos y sus proveedores al momento de realizar un pedido, creando una alternativa basándose en sus necesidades, para que estos no solo puedan comunicarse entre sí, sino que también puedan comercializar sus productos de una manera más eficiente, de tal modo que se permita reducir el desabastecimiento en las tiendas de barrio y se logre aportar a la economía del país.

Palabras Clave:

- Tecnología
- Comunicación
- Desabastecimiento
- Arquitectura de software
- Scrum

Abstract

Technology is a tool that turns out to be extremely important today, which allows us to communicate with other people and make ourselves known in order to generate a greater impact on a target audience, by working together with existing advertising or marketing strategies. Technology can also become fundamental to driving or controlling a business, as it can increase its productivity and efficiency by breaking down the barrier between a supplier and its customer to communicate at a distance. In Colombia, companies have been implementing digital strategies very slowly compared to other countries [1]. In fact, COVID-19 and the national strike were triggers that forced businesses, such as neighborhood stores and their suppliers, to adapt to a new digital age in which they can sell, buy or distribute their products more efficiently. However, given the economic crisis, the lack of resources, communication and the shortage of supplies that were going through at that time, it has been difficult to generate a digital transition in many neighborhood stores and their suppliers that require it to boost their business. Since, although various mobile or web applications have been developed that seek to serve as intermediaries, they do not cover the needs of most people or businesses.

This thesis will seek to create a technological solution by means of a Chatbot that allows to improve communication between shopkeepers and their suppliers at the time of placing an order, creating an alternative based on their needs, so that they can not only communicate with each other, but also market their products in a more efficient way, so that it is possible to reduce shortages in local shops and contribute to the economy of the country.

Keywords:

- Technology
- Communication
- Shortage of supply
- Software architecture
- Scrum

Índice general

Índice de figuras

Capítulo 1

Descripción del Problema

1.1. Planteamiento del Problema

A pesar de la reactivación económica que se ha venido dando en el país después de un largo periodo de pandemia y de los efectos del paro nacional que han afectado directamente al pequeño comercio,un problema que persiste es que las tiendas de barrio aún no se han podido recuperar económicamente. Según la Federación Nacional de Comerciantes (FENALCO), se tiene estimado que más de 11.000 tiendas de barrio pueden desaparecer [2], lo cual puede llegar a afectar más de un millón de familias y cerca de 1.75 millones de empleos que por lo general dependen de estos negocios, lo cual perjudica la economía del país, debido a su disminución en cuanto a productividad y aporte al desarrollo económico.

De acuerdo con los tenderos que participaron en la encuesta realizada por FENALCO, algunos de los obstáculos a los cuales se han enfrentado los tenderos son la poca financiación que han obtenido por parte del gobierno, la falta de apoyo o comunicación con los proveedores, los altos costos por parte de sus intermediarios y el atraso tecnológico con el que cuentan. Por esta razón, poco a poco este gremio ha buscado acoger algunas iniciativas digitales que permitan evitar el cierre de algunos establecimientos que resultan ser claves para poder abastecerse [3]. Asimismo, se debe tener en cuenta que se ha generado un aumento en los precios de los productos debido a que a muchos proveedores de las tiendas de barrio, se les ha dificultado distribuir o elaborar los mismos. Es por esto, por lo que aún la mayoría de estos negocios requieren cierto tipo de apoyo que les permita realizar un cambio o transformación en su negocio con el fin de poder mantenerlos a flote, logrando tener una mayor oferta digital y una mejor comunicación con sus clientes.

Una de las grandes dificultades con la que se enfrentan los proveedores y las tiendas de barrio, es la comunicación, ya que esta resulta ser una herramienta clave tanto para el dueño del negocio como para sus clientes potenciales; dado que si se llega a dar algún caso en el que el tiempo de espera para los clientes sea muy largo, si no se les atiende sus llamadas, si no se les resuelven sus preguntas frecuentes, si el personal no está disponible en el momento o incluso si no se toma de manera adecuada la solicitud del cliente; esto, representará una pérdida de oportunidades que se podrá ver reflejada a futuro en la disminución de sus ventas, la pérdida de confianza de los clientes o el posible fracaso de un proyecto importante. Además, a medida que esto se repita, los clientes perderán poco a poco su confianza y vínculos con la empresa, debido al poco interés que se les demuestra por resolver sus problemas, lo que conlleva a que estos busquen soluciones en otra parte.

Un estudio reciente por parte de Microsoft Colombia ha demostrado como muchas pequeñas empresas han logrado salir adelante con la implementación de nuevas tecnologías en sus negocios [4]. Entre estas tecnologías, se encuentra el desarrollo de páginas web o aplicaciones móviles, las cuales

han permitido generar cierto comercio electrónico (e-commerce) para vender sus productos a través del internet. Mientras que, por otra parte, se tiene el comercio conversacional (c-commerce), el cual suele utilizar herramientas como lo son los asistentes virtuales (Chatbots) en las aplicaciones de mensajería instantánea, para poder realizar el proceso de compra/venta o resolución de preguntas frecuentes de manera automatizada. Sin embargo, un problema clave con el que se enfrenta muchas veces la comunicación de estos negocios, está relacionado con el diseño e interfaz de estas tecnologías, ya que, si el uso de esta resulta ser complicado o poco intuitivo para el usuario, éste optará por no utilizarla.

A pesar de la evolución de los grandes supermercados, las tiendas de barrio siguen siendo el más importante medio de distribución y abastecimiento de los productos de la canasta familiar para los Colombianos [5]. Y aunque en la actualidad existen algunas tecnologías como SurtiApp[6] y Chiper[7] que han intentado aportar soluciones para establecer una comunicación directa entre proveedores y tiendas, no brindan un servicio personalizado y adaptado a las necesidades de la tienda de barrio típica colombiana por los motivos que se describen a continuación.

Las soluciones existentes no tienen en cuenta que en las tiendas de barrio trabajan, en promedio, entre 1-3 personas [8] y éstas no cuentan con los recursos necesarios para acceder a un plan de datos postpago en su celular, mediante el cual puedan acceder a una aplicación web o móvil que se haya desarrollado.

Adicionalmente, según el informe de Fenaltiendas (Programa Social de Apoyo al Tendero de Barrio), creado por la Federación Nacional de Comerciantes (FENALCO) hace más de 21 años, la edad del tendero promedio se encuentra entre los 43 y 44 años [9], lo cual refuerza el hecho de que esta labor la realizan principalmente personas mayores de edad que quizás no se encuentran muy familiarizadas con el uso de nuevas tecnologías, las cuales por lo general no cuentan con internet en su tiendas y están acostumbradas a realizar todo manualmente, Es por esto, que este público objetivo como lo demuestra la Asociación Colombiana de Empresas de Investigación de Mercados y Opinión Pública (ACEI) [10], por lo general solo hace uso de algunos canales de mensajería instantánea y redes sociales como lo son Facebook, Whatsapp e Instagram. Lo cual es muy importante tener en cuenta en el momento de pensar en una solución para sus problemas de comunicación.

Por otro lado, se tiene que, los chatbots existentes que se pueden generar en plataformas como Twilio[11] y Cliengo[12], aunque permitirían mejorar la comunicación entre los proveedores y las tiendas de barrio, no cuentan con una interfaz dentro de sus bots que sea fácil de utilizar por los tenderos, la cual permita realizar un pedido a un proveedor que cuente con distintos productos y categorías de una manera efectiva. Actualmente algunos canales de chat han venido experimentando e implementando Apis en beta que permiten combinar dentro de su canal aplicaciones web con los bots tradicionales, lo cual permite explorar una nueva herramienta innovadora para los chatbots en la cual se pueda diseñar una interfaz que se ajuste a la necesidades planteadas anteriormente.

La implementación de un Chatbot que cuente con una interfaz gráfica intuitiva en una plataforma de mensajería instantánea resulta ser un reto llamativo, el cual buscará dar solución a los problemas mencionados anteriormente, dado que este se podrá adaptar a su modelo de negocio y a las necesidades del tendero. Además, este resultará ser mucho más rentable y fácil de utilizar por

cualquier persona, ya que estarán utilizando un software con el que ya se encuentran familiarizados. El cual, con la ayuda de la ingeniería de software, permitirá mejorar la comunicación entre los distintos proveedores y tenderos, teniendo en cuenta sus necesidades y su cadena de suministros para poder dejar a un lado los intermediarios.

1.1.1. Formulación

¿Cómo diseñar e implementar un chatbot que permita mejorar la comunicación entre los proveedores y tiendas de barrio?

1.1.2. Sistematización

- ¿Cómo identificar las necesidades de los *stakeholders*?
- ¿Cómo diseñar un chatbot que satisfaga las necesidades de los stakeholders?
- ¿Cómo se implementará el chatbot?
- ¿Cómo validar los requisitos del chatbot desde el punto de vista funcional y de usabilidad?

1.2. Objetivos

1.2.1. Objetivo General

Diseñar e implementar el prototipo de un chatbot que permita mejorar la comunicación entre proveedores y tiendas de barrio.

1.2.2. Objetivos Específicos

- Revisar las necesidades de los stakeholders para poder establecer los requisitos funcionales y no funcionales que se requieren.
- Diseñar la arquitectura, modelo de datos y elegir las herramientas que se utilizarán para desarrollar el chatbot.
- Desarrollar el chatbot teniendo en cuenta los requisitos y diseños planteados para este proyecto.
- Validar el chatbot desde el punto de vista funcional y de usabilidad, cumpliendo e implementando los requisitos funcionales de acuerdo con lo planteado.

1.3. Justificación

De acuerdo con las cifras obtenidas por el boletín del primer trimestre del año 2022 a cargo del Ministerio de Tecnologías de la Información y las Comunicaciones (MinTIC)[13]. Se puede apreciar que Colombia cuenta con alrededor de 76 millones de líneas de telefonía móvil actualmente, de las cuales solo 18,2 millones de usuarios cuentan con un plan postpago, mientras que los 57,8 millones restantes cuentan con un plan prepago. Esto, considerando que la mayoría de usuarios no son de estratos socioeconómicos altos o prefieren realizar recargas mínimas mensuales que les permita tener acceso a ciertas aplicaciones de mensajería instantánea para comunicarse, ya que estas no consumen datos, funcionan con conexión a alguna red wifi o incluso se dan gratis con la misma recarga.

Sumado a esto, según las estadísticas de eMarketer [14], el uso de dispositivos móviles se ha vuelto esencial, ya que en primer lugar, alrededor del 45 % de las compras de productos en línea se realizan a través de estos dispositivos y en segundo lugar, cerca del 94.8 % de usuarios entre los 16 y 64 años de edad que acceden a internet, lo hacen por medio de estos sistemas y pasan un promedio de 5 horas al día interactuando con ellos. Por lo cual, si se llega a proponer una aplicación web o móvil como solución del problema planteado, dicha solución no abarcaría en su totalidad el público objetivo, ya que si los tenderos no cuentan con un plan de datos, no podrían descargar o acceder a la aplicación creada, ni hacer uso de la misma, lo cual sería ineficiente.

El proceso que realizan los proveedores para poder promocionar o vender sus productos a los tenderos, resulta ser muy tedioso ya que estos por lo general le pagan a un intermediario para que envíe a ciertos vendedores con una cartilla física de sus productos, como se puede evidenciar en la Figura 1.1, y que estos vayan de tienda en tienda a los barrios tomando los pedidos de cada uno de los tenderos, lo cual toma mucho tiempo para ambas partes e incluso no es competente, ya que este trabajo se podría reemplazar con una tecnología que permita agilizar el proceso y que mejore la comunicación entre ellos.

Figura 1.1: Cartilla de productos, tomada de: Distri Romel S.A.

La implementación de un chatbot para los proveedores de las tiendas de barrio puede ser de gran ayuda, ya que estos podrán interactuar de manera constante con los tenderos, las 24 horas del día, los 7 días a la semana, lo cual permitirá reducir el desabastecimiento a través de una buena

comunicación, mientras mejoran continuamente la calidad de sus respuestas al cliente y logran mantener bajos costos para la empresa; dado que un chatbot cuenta con distintos casos de uso para los cuales puede ser implementado, entre ellos los más comunes y que aportan mayor valor a la empresa se encuentra la atención al cliente, su incorporación en un equipo de ventas y el comercio electrónico [15].

Entre las principales ventajas de un chatbot se encuentra la disminución de la carga operativa para los empleados, ya que este permite eliminar tareas tediosas que lleguen a consumir mucho tiempo, como lo pueden ser las preguntas frecuentes en búsqueda de alguna información o servicio de la empresa. Lo cual, permitirá a los empleados ejercer tareas más significativas y brindarle una mayor comodidad al cliente como se puede evidenciar en las encuestas de Statista realizadas a los encargados del servicio al cliente, en la cual comparten su opinión sobre la implementación de los chatbots. Obteniendo un 72 %, de empleados que al trabajar en tareas más complejas sienten que están mejorando sus habilidades y generando un mayor impacto dentro de la compañía, llegando incluso a incrementar la satisfacción y productividad del 59 % de los mismos [16].

La personalización y mejora en tiempos de respuesta que llegan a brindar los chatbots ha resultado ser un éxito rotundo, ya que después de todo, el marketing personalizado y un buen servicio al cliente garantizan buenos resultados. De acuerdo con encuestas realizadas por Accenture [17] y Epsilon [18], se reveló que un 91 % de los consumidores eligen con mayor probabilidad marcas que ofrecen ofertas y recomendaciones personalizadas, obteniendo también que, es más probable que el 80 % de estos realicen una compra si cuentan con una experiencia personalizada, como la que puede brindar un chatbot [19].

Muchas empresas que residen en Colombia como Tuya, Sura, Banco Falabella, Banco de Bogotá y Bancolombia se han visto beneficiadas con la implementación de los chatbots, dado que estos les han permitido gestionar las principales dudas o interacciones de los usuarios, de una manera más ágil y al alcance de todos. Esto, gracias a la evolución de la inteligencia artificial, la cual permite incluso llegar a simular con gran realismo ciertas conversaciones complejas [20]. Es tanto el valor que puede aportar a una empresa un chatbot que se estima incluso que los ahorros de costos por el uso de estos en la industria bancaria a nivel mundial para 2023 según Juniper Research será de alrededor de $ 7.3 billones, a medida que evolucione la experiencia automatizada del cliente [21].

Por lo tanto, con los avances que se tienen actualmente y los beneficios nombrados, se puede apreciar que es fundamental integrar las herramientas digitales, como lo son los chatbots, a las tiendas de barrio para poder mejorar la comunicación o interacción con sus proveedores, para así garantizar una experiencia personalizada que sea fácil de usar y genere valor para ambas partes.

1.4. Delimitaciones y Alcances

- La población objetivo será la ciudad de Cali.
- El sistema a desarrollar será implementado y puesto a prueba con un proveedor de la misma ciudad.
- Se hará uso de la plataforma de mensajería Telegram debido al crecimiento que ha tenido el

canal actualmente como se puede evidenciar en [22] y a la facilidad de uso de sus API, ya que a diferencia de Whatsapp / Meta, Telegram no solicita la verificación de la empresa para trabajar con sus API.

1.4.1. Entregables

Una vez finalizado el proyecto de grado se tiene como propósito entregar los siguientes ítems:

- Código fuente del proyecto.
- Documentación relacionada con el software.
- Documento de trabajo de grado.

Capítulo 2

Desarrollo del Proyecto

2.1. Marco de Referencia

2.1.1. Áreas Temáticas

Las areas temáticas del proyecto son:

- Software and its engineering →Software creation and management →Designing software →Requirements analysis.
- Software and its engineering →Software creation and management →Designing software →Software design engineering.
- Software and its engineering →Software creation and management →Software development process management →Software development methods →Agile software development.
- Computing methodologies →Artificial intelligence →Natural language processing →Information extraction.
- Software and its engineering →Software creation and management →Software verification and validation →Software defect analysis →Software testing and debugging.

Se pueden encontrar las diferentes áreas temáticas, de acuerdo con las categorías de ACM en el siguiente enlace: (http://www.acm.org/about/class/ccs98-html).

2.1.2. Marco Teórico

Para lograr desarrollar el proyecto que se está planteando, es necesario tener conocimiento de todos los conceptos que se van a tratar, entre ellos se encuentran los que abarca la ingeniería de software, las metodologías ágiles para el desarrollo de software, la arquitectura de software, el diseño de este, tecnologías, frameworks que se utilizarán, etc. Por lo cual se mencionan algunos de estos conceptos clave que se deben tener en cuenta a continuación:

- **Metodologías ágiles:** los métodos ágiles universalmente dependen de un enfoque iterativo para la especificación, desarrollo y entrega de software, y principalmente fueron diseñados para apoyar al desarrollo de aplicaciones de negocio donde los requerimientos del sistema con normalidad cambian rápidamente durante el proceso de desarrollo. Están pensados para entregar el software funcional de forma rápida a los clientes, quienes pueden entonces proponer

que se incluyan en iteraciones posteriores del sistema nuevos requerimientos en los mismos [23]. Además, el uso de este tipo de metodologías, permiten no solo mejorar la calidad del producto que se pretende entregar al final, sino que también estimulan el trabajo colectivo, permiten predecir resultados de una manera más fácil, minimizar los riesgos del proyecto, reducir costos e incluso llegar a brindar una mayor satisfacción al cliente [24].

- **Scrum:** Es una de las metodologías ágiles más utilizadas en el mercado, se caracteriza por ser incremental e iterativa. Busca siempre estar en contacto con el cliente permitiendo la flexibilidad, efectividad y la comunicación constante durante el desarrollo. La planificación detallada se realiza sobre cortos espacios de tiempo (sprints), lo que permite una constante retroalimentación, que proporciona inspecciones simples y un ciclo de vida adaptable [25].Esta cuenta con ciertos pasos y roles que se deben cumplir en el proceso para poder lograr un objetivo en común, como se describen a continuación:

Figura 2.1: Metodologia scrum: Fases de un sprint [26]

- **Design thinking:** Es una metodología de resolución de problemas adaptado para la investigación de problemas débilmente definidos, centrado en las personas, centrando en las posibilidades e impulsada por hipótesis de valor [27].Esta metodología cuenta con 5 etapas importantes que son:

  - **Descubrimiento:** se define un reto (desafío), en este caso un problema de investigación

específica o intencional, al que se le denomina un desafío de diseño. El desafío debe ser abordable, comprensible, realizable y estar bien delimitado.

- **Interpretación:** se busca generar conocimientos significativos a partir de lo que se observa, de las visitas de campo o una simple conversación con los objetos de estudio, pares o asesores. No es una tarea fácil ya que implica ordenar y juntar pensamientos hasta encontrar un punto de vista convincente y una clara orientación para la "ideación" o generación de ideas que posteriormente se reflejarán en un modelo de investigación.
- **Ideación:** En esta etapa se desarrollan, no solo las ideas, sino que también se busca obtener una propuesta para generar el modelo de investigación, consolidar los objetivos generales y los específicos, así como las hipótesis de trabajo. Asimismo, se realiza una revisión de qué tan viable es la propuesta de investigación y la actualidad del tema elegido.
- **Experimentación:** se construyen prototipos o modelos para hacer tangibles las ideas con el fin de poder aprender mientras se construyen y se comparten con otras personas; es una forma de aprender a través de la retroalimentación de otros, para poder seguir mejorando y refinando, las ideas o modelos de investigación.
- **Evolución:** se define como el desarrollo del modelo de investigación. Incluyendo la planificación de los próximos pasos, la comunicación de la idea y documentación del proceso. Es una oportunidad para poder planificar los próximos pasos en el proceso de investigación, socializar el tema y con ello construir comunidad con investigadores con temas afines.

- **Principios de usabilidad:** La usabilidad se refiere a la simplicidad con las que las personas interactúan con una herramienta para alcanzar un objetivo en concreto. En cuanto al desarrollo web, es la encargada de describir en qué medida una aplicación web resulta fácil de usar para una persona, ya que, si esto se garantiza, los usuarios podrán tener una mejor experiencia para alcanzar sus objetivos dentro de la misma.

  Los principios de usabilidad, también llamados principios heurísticos, son una serie de 10 ideales y fundamentos planteados por Jakob Nielsen, una de las personas más respetadas en el ámbito mundial en cuanto a usabilidad web, estos principios permiten desarrollar productos que al tener en cuenta las necesidades y el comportamiento de los usuarios, logra tener un mayor grado de acogida entre ellos [28],[29]. Estos principios son:

  - **Visibilidad del estado del sistema:** Siempre se debe mantener informado al usuario de lo que está sucediendo en la web y hay que brindarle una respuesta en el menor tiempo posible.
  - **Relación entre el sistema y el mundo real:** Es importante que la página web o la aplicación utilice el idioma que el usuario entiende, incluyendo expresiones y vocabulario que le resulten familiares. La información presentada también debe seguir un orden coherente y debe ser fácil de comprender para el usuario.

- **Control y libertad del usuario:** En caso de realizar alguna opción por error. El usuario debe poder deshacer o repetir una acción previamente realizada.
- **Consistencia y estándares:** Es importante establecer convenciones lógicas y mantenerlas siempre. Los usuarios no deberían tener que preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. Se recomienda seguir las convenciones de la plataforma y la industria.
- **Prevención de errores:** Diseñar cuidadosamente las interfaces para lograr prevenir cualquier error que pueda cometer el usuario.
- **Reconocimiento antes que recuerdo:** Es importante que en el diseño de un sitio web se muestren claramente las opciones y acciones disponibles, de manera que el usuario no tenga que recordar información entre distintas secciones o partes de este.
- **Flexibilidad y eficiencia de uso:** El sitio web debe estar preparado para todo tipo de usuario, desde los más novatos hasta los más experimentados.
- **Diseño estético y minimalista:** Es importante evitar incluir información irrelevante en las páginas web para no distraer al usuario y evitar que tenga una experiencia negativa durante su navegación.
- **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores:** Se debe buscar que todos los mensajes de error que puedan aparecer estén expresados en un lenguaje entendible por todos, no por códigos.
- **Ayuda y documentación:** Aunque estos principios intentan siempre que el usuario no tenga que recurrir a un tipo de ayuda o documentación, se recomienda brindar ciertas ayudas a los usuarios para que estos puedan llevar a cabo sus tareas.

- **Técnica de recolección de la información enfoque cualitativo:** Recolección de datos, pero sin utilizar la medición numérica y el análisis estadístico. Interpreta la información utilizando instrumentos específicos, con el fin de establecer preguntas de investigación, hipótesis, generar teorías y conocimientos nuevos o novedosos durante el proceso de interpretación. Entre estas técnicas de recolección de información se encuentran la observación, las entrevistas, las técnicas proyectivas y los grupos focales [30].

- **Estructura de Software y Arquitectura:** Se refiere a la manera en la que se encuentran organizados los componentes que modelan un programa informático denominado como software, haciendo uso de distintos patrones o guías que ayudan a los desarrolladores, analistas y demás cargos relacionados, a cumplir con los requerimientos de una aplicación, dividiendo el software en partes lógicas y funcionales que se relacionan entre sí [31].

  - **Patrón de diseño:** son modelos que funcionan como una guía para dar soluciones a problemas comunes que se pueden presentar en el desarrollo de software y otros ámbitos relacionados como el diseño de interacción o interfaces [32].

  - **Diseño de interfaz de usuario:** Se encarga de garantizar una efectiva interacción entre el humano y la máquina, manteniendo unos principios como el aprendizaje, la familiaridad del usuario y la consistencia, entre otros.

- **API (Application Programming Interfaces):** Es un conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas [33].

- **Chatbot:** es un programa informático que permite procesar y simular conversaciones humanas usando el procesamiento del lenguaje natural, ya sea de manera escrita o hablada, para permitir a los humanos interactuar con dispositivos digitales como si se estuvieran comunicando con una persona real con disponibilidad 24/7. Es una tecnología que permite al usuario mantener una conversación a través de un software que puede llegar a integrarse en algún sistema de mensajería determinado, como, por ejemplo: Facebook, Whatsapp, Telegram, etc. Los chatbots pueden ser tan simples como programas que responden a consultas sencillas con una sola línea de texto o tan complejos como lo son los asistentes digitales que pueden llegar a evolucionar y aprender de manera personalizada, a medida que van reuniendo y procesando mayor cantidad de información [34].

- **Webhook:** Es un mecanismo de comunicación en el desarrollo de aplicaciones web que se basa en el protocolo HTTP para que dos interfaces de programación puedan comunicarse de manera automatizada, permitiendo enviar datos en tiempo real a otra aplicación cuando ocurre un evento específico. Asimismo, un webhook permite ir recibiendo los datos de manera asincrónica de otra aplicación cuando sea necesario generalmente mediante un método POST a la URL predefinida [35].

- **Comercio Conversacional (C-Commerce):** Hace referencia a cualquier clase de conversación que se puede llegar a tener en tiempo real entre los distintos clientes y marcas, con el fin de adquirir o vender, tanto un producto, como un servicio, con el fin de llevar a cabo una transacción; haciendo uso de herramientas digitales como las aplicaciones de mensajería instantánea, ya sea mediante chatbots, inteligencia artificial o personas reales. Lo cual permite tener ciertas ventajas tales como brindar un servicio más eficaz al cliente por medio de la automatización de las conversaciones, minimizar ciertas barreras que puedan existir en el momento de efectuar una compra e incluso llegar a influir en las distintas audiencias que pueden formar parte del mercado objetivo [36].

Teniendo en cuenta lo planteado anteriormente se utilizará como metodología de desarrollo ágil, la metodología Scrum, para el cual se proyectarán 6 sprints, en donde cada sprint será un trabajo mensual con sus respectivas reuniones y pruebas. Además, se hará uso de la ingeniería de software para realizar un levantamiento de requerimientos en el momento en que se recolecta la información por medio de entrevistas a los distintos stakeholders.

- **Kits de desarrollo:**

- **React.js:** Es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Es mantenido por Facebook y la comunidad de software libre. Ver [37]
- **Flask:** es un framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código. Está basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD. Ver [38]
- **Telegram:** Es una popular plataforma de mensajería y VOIP (Voz sobre protocolo de internet), La aplicación está enfocada en la mensajería instantánea, el envío de varios archivos y la comunicación en masa. Es muy utilizada ya que ofrece algunas funciones mejoradas en cuanto a privacidad y cifrado de mensajes, así como soporte para funciones de grandes grupos de chat y es multiplataforma. Ver [39]

2.1.3. Trabajos Relacionados

- **Surtiapp:** Es una plataforma tecnológica gratuita que fue desarrollada en Colombia y funciona como intermediaria prestando servicios únicamente de abastecimiento a tiendas, distribuyendo productos para el hogar perecederos y no perecederos. Se puede ver que ofrecen un servicio a domicilio a través de su aplicación web y móvil. El software planteado en este trabajo de grado se diferencia de "Surtiapp" debido a que cuenta con un enfoque distinto, el cual busca apoyar a distintas tiendas de barrio y a sus proveedores, con el fin de brindar un apoyo a estos sectores, para así lograr reducir el desabastecimiento y aportar a la economía de nuestro país [6].

- **Chiper:** Es una aplicación móvil gratuita que fue desarrollada en Colombia y funciona como intermediaria prestando el servicio de abastecimiento a tiendas y generando empleo a aquellas personas que estén interesadas en generar ingresos ayudando a distribuir sus productos en sus domicilios. Al igual que "Surtiapp", cuenta con un enfoque distinto y distribuyen los mismos tipos de productos para el hogar. El software planteado en este trabajo de grado se diferencia de "Chiper" y de "Surtiapp", dado que se busca a través del procesamiento natural del lenguaje crear un chatbot que funcione como un C-commerce con el fin de poder facilitar la comunicación e interacción entre los distintos stakeholders al brindar estos servicios [7].

- **Twilio:** Es una una plataforma de comunicaciones desarrollada en San Francisco, California, la cual permite desarrollar aplicaciones que hagan y reciban llamadas, mensajes de texto, elaboren funciones de comunicación y registro, usando APIs, propias del servicio web; con el fin de ayudar a las empresas a vender más gracias al internet, implementando entre sus tecnologías los chatbots.Una gran diferencia con el software que se quiere implementar en este trabajo de grado, es que sus chatbots no cuentan con una interfaz grafica integrada que resulte ser intuitiva para cualquier usuario y que además permita que estos realicen su pedido a través de ella [11].

- **Cliengo:** Es una plataforma tecnológica que fue desarrollada en Argentina, la cual brinda soluciones digitales a sus clientes a través de los chatbots que estos manejan, para distintas aplicaciones de mensajería instantánea como Whatsapp, Facebook e Instagram, siguiendo un enfoque parecido al que se requiere para este trabajo de grado, pero con la diferencia de que estos se basan en un marketing B2C (Business to Consumer), cuando lo que se necesita realmente es un marketing B2B (Business to Business), el cual tenga en cuenta la experiencia que se le brinda al usuario y la cadena de suministro que existe entre las tiendas de barrio y sus proveedores. Asimismo, al igual que "Surtiapp", sus chatbots no cuentan con una interfaz gráfica intuitiva para el usuario[12].

2.2. Metodología

2.2.1. Tipo de Estudio

La metodología de investigación del proyecto es exploratoria, debido a que el éxito de este depende de la comodidad, usabilidad y satisfacción por parte de los stakeholders en el momento en que este sea puesto a prueba. Lo cual, brindará una alternativa innovadora que teniendo en cuenta los requerimientos, permitirá mejorar la comunicación de los proveedores y las tiendas de barrio, logrando reducir el desabastecimiento e impulsando la economía del país.

2.2.2. Actividades

- **Objetivo 1:** Identificar plataformas o aplicaciones con enfoques similares, para revisar las necesidades de los diferentes sectores haciendo uso de la ingeniería de software para poder establecer los requisitos funcionales y no funcionales que requiere el proyecto.
  - Buscar e identificar las diferentes plataformas o aplicaciones que poseen características similares a la propuesta planteada.
  - Realizar consultas e investigaciones pertinentes para verificar procesos de las funciones que realizan dichas plataformas o aplicaciones.
- **Objetivo 2:** Definir las estrategias y protocolos para lograr mejorar la comunicación entre los stakeholders de la solución tecnológica planteada.
  - Analizar los modelos y patrones usados para garantizar una mejor comunicación entre los stakeholders.
  - Identificar y extraer toda la información relevante del modelo seleccionado para el desarrollo del proyecto.
  - Identificar y extraer los requisitos necesarios para el desarrollo del proyecto.
  - Identificar y seleccionar las métricas para la evaluación del progreso de este.
- **Objetivo 3:** Diseñar los diferentes componentes que se utilizarán.

  - Diseñar la arquitectura de software que mejor se acomode a los requisitos planteados.
  - Diseñar un modelo de datos que recoja las entidades y atributos planteados en los requisitos.

- **Objetivo 4:** Desarrollar la solución tecnológica para que se ajuste a los requisitos planteados para este proyecto.

  - Implementar los diseños previos teniendo en cuenta los requisitos establecidos.
  - Desarrollar el modelo de datos.
  - Desarrollar el código correspondiente.
  - Integración del proyecto

- **Objetivo 5:** Analizar los resultados del proyecto

  - Realizar pruebas de usuario pertinentes.
  - Realizar encuestas de satisfacción a los stakeholders.
  - Realizar reuniones.
  - Realizar correcciones y ajustes.

2.3. Resultados Esperados

- Documento de trabajo de grado que incluye todos los pasos del diseño, desarrollo y pruebas del software.

- Chatbot en su fase beta de desarrollo.

2.4. Cronograma

Cronograma por semana incluyendo las actividades descritas en la Sección 2.2.2.

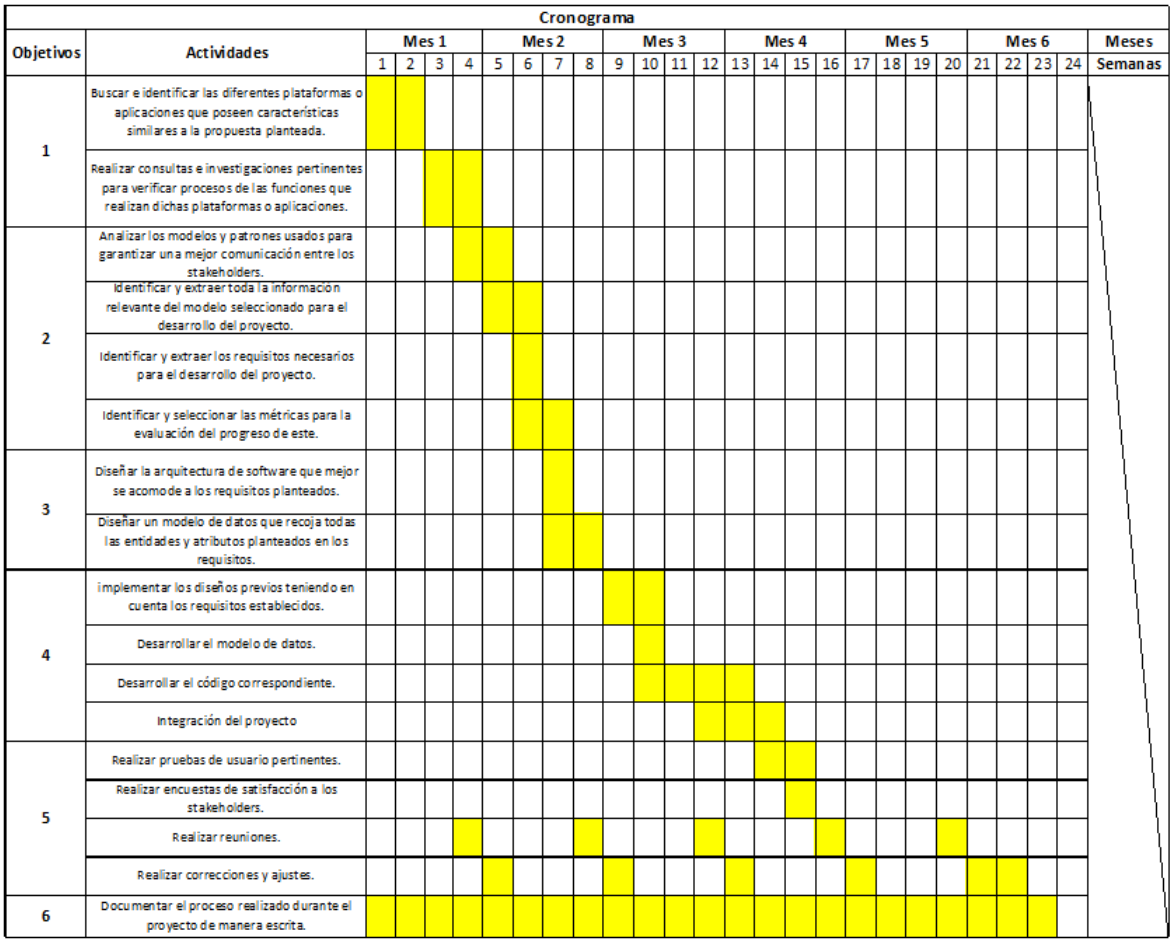


Figura 2.2: Cronograma de actividades del proyecto.

2.5. Recursos

2.5.1. Humanos

- Juan Pablo Garcia (Director de Proyecto de Grado), Director Programa de Innovación por Diseño red SUGAR.
  - Ingeniero de Sistemas y Computación. Pontificia Universidad Javeriana. Cali, 2008.
  - ME310 Global Team-Based Design Innovation with Corporate Partners - Design Thinking & Entrepreneurship. Stanford, 2008.
  - Magíster en Administración. Universidad Icesi. Cali, 2013.
  - Diplomado Educación en Ingeniería. Pontificia Universidad Javeriana. Cali, 2016.
- Julian Paredes Conde: Estudiante de pregrado de la Pontificia Universidad Javeriana Cali.

2.5.2. Técnicos

Equipos y materiales necesarios para la realización del proyecto:

- Zoom: Plataforma de videoconferencias para realizar las reuniones. Brindado por la Universidad Javeriana Cali.
- Computador proporcionado por el autor del proyecto.
- Acceso a internet.
- Biblioteca de la universidad Javeriana Cali.

2.5.3. Presupuesto

Presupuesto que incluye los costos de los recursos humanos y técnicos para llevar a cabo el proyecto:

SMMLV Colombia
$ 1.000.000

Figura 2.3: SMMLV 2022

Costo total del proyecto
$ 37.428.000

Figura 2.4: Costo total del proyecto

Recursos Humanos	Cantidad SMMLV	Costo mensual	Tiempo estimado (en meses)	Subtotal
Juan Pablo Garcia	5	$ 5.000.000	6	$ 30.000.000
Julian Paredes Conde	1	$ 1.000.000	6	$ 6.000.000
Total				$ 36.000.000

Figura 2.5: Costo de recursos Humanos

Recursos Fisicos	Costo Mensual	Meses	Subtotal
Internet	$ 98.000	6	$ 588.000
Energia electrica	$ 140.000	6	$ 840.000
Total			$ 1.428.000

Figura 2.6: Costo de recursos Fisicos

Capítulo 3

Obtención y Análisis de Requisitos

3.1. Estrategias Implementadas para la Educción de Requisitos

- Lluvia de ideas: Se llevó a cabo una sesión de lluvia de ideas, con el fin de generar una serie de preguntas que permitieran identificar los requisitos necesarios. Estas preguntas se plantearon a los distintos stakeholders, quienes brindaron explicaciones, respuestas o sugerencias.

- Casos de Uso: Se utilizó un diagrama de casos de uso (Véase en la Figura 3.1) para examinar las distintas acciones que pueden realizar los usuarios y, a partir de esto, se obtuvieron más requisitos que permiten modelar de manera más precisa el comportamiento de los distintos roles que conforman el sistema.

- Prototipado: Se desarrollaron varios prototipos utilizando la herramienta "Balsamiq-mockups", con el objetivo de identificar las funcionalidades que no se habían contemplado y, de esta manera, poder modelar con mayor exactitud el comportamiento del Chatbot.

3.2. Descripción general del Chatbot

El objetivo del Chatbot es mejorar la comunicación y la relación entre los proveedores y las tiendas de barrio, a través de diversas funcionalidades que conectan a ambos tipos de usuarios. Para lograr esto, el Chatbot implementa distintas estrategias para identificar los requisitos necesarios y tener una visión más clara de cómo debe funcionar. Cabe destacar que el Chatbot funciona mediante dos perfiles previamente definidos, que serán: 1. Usuario tendero, quien podrá solicitar información de contacto, ayuda e incluso registrarse, para posteriormente realizar un pedido a un proveedor. 2. Usuario proveedor, quien podrá visualizar los productos solicitados por los tenderos para así poder despachar su pedido.

El Chatbot debe permitir a un tendero realizar un pedido de manera correcta a un proveedor, lo cual implica que este debe contar con una interfaz intuitiva que permita al tendero interactuar entre las distintas acciones disponibles. Del mismo modo, el Chatbot debe permitir una administración sencilla de los datos de entrada para su regulación y visualización por parte del proveedor, lo que implica que el mismo debe contar con una herramienta que permita al proveedor observar y/o modificar los distintos datos presentes.

3.3. Diagrama de casos de uso

Este diagrama muestra el comportamiento de todos los objetos involucrados, su estructura y relaciones entre sí. A través de este se pretende representar el funcionamiento del chatbot para los usuarios tenderos y el proveedor.

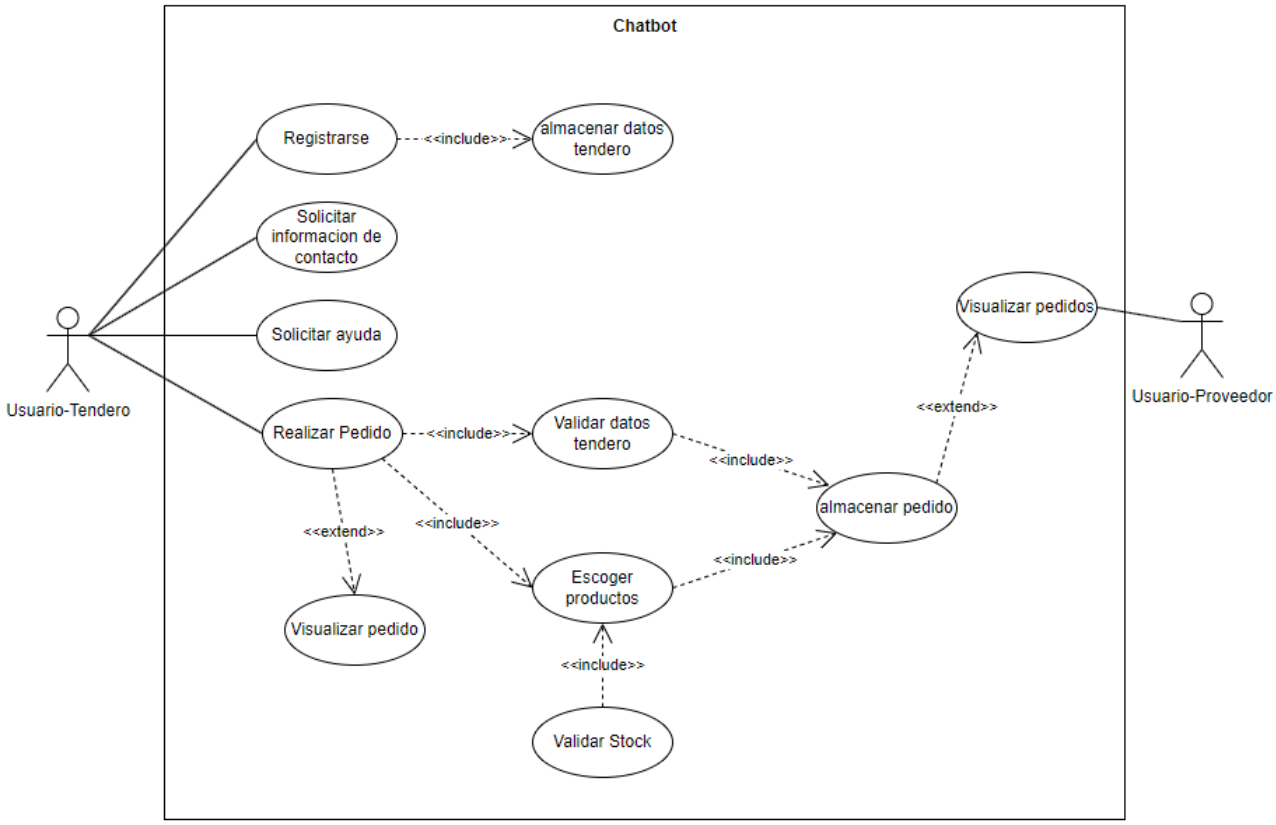


Figura 3.1: Diagrama de casos de uso.

3.4. Requisitos

Con el propósito de establecer las pautas y normas para el correcto funcionamiento del "Chatbot para mejorar la comunicación entre tiendas de barrio y sus proveedores", se establecieron los requisitos funcionales y no funcionales, obtenidos a través del uso de diversas técnicas de educción de requisitos, planteadas anteriormente en el marco teórico del presente documento. Para ver al detalle los requisitos planteados, es necesario revisar la sección 10.1 del capítulo de Anexos.

Capítulo 4

Diseño del Software

4.1. Introducción

Este capítulo explicará el diseño del software construido para este proyecto. Empezando con un diagrama de flujo, el cual permite representar y visualizar una secuencia de pasos o actividades que se llevan a cabo en el proceso de la realización de un pedido desde la perspectiva de un tendero. Siguiendo con la arquitectura del chatbot, pues esta representa una parte fundamental del mismo; argumentando la escogencia de su tipo. Continuando con el diagrama de la base datos, el uso que se le dará en el software, su naturaleza y, por qué fue escogida para este proyecto. Para finalmente mostrar los prototipos realizados que se tendrán en cuenta para el desarrollo del Chatbot.

4.2. Diagrama de Flujo:

A continuación, en la Figura 4.1, se observa el diagrama de flujo mediante el cual se pretende comprender, modelar o reflejar, de manera visual, una serie de acciones o tareas que se realizan para completar, en este caso, la realización de un pedido a un proveedor por parte de un tendero mediante una conversación que se lleva a cabo con el Chatbot planteado en Telegram. Este tipo de diagramas se emplean para representar y examinar de forma más precisa y sencilla los procedimientos existentes, lo que facilita la identificación de dificultades y oportunidades de mejora en el flujo de trabajo.

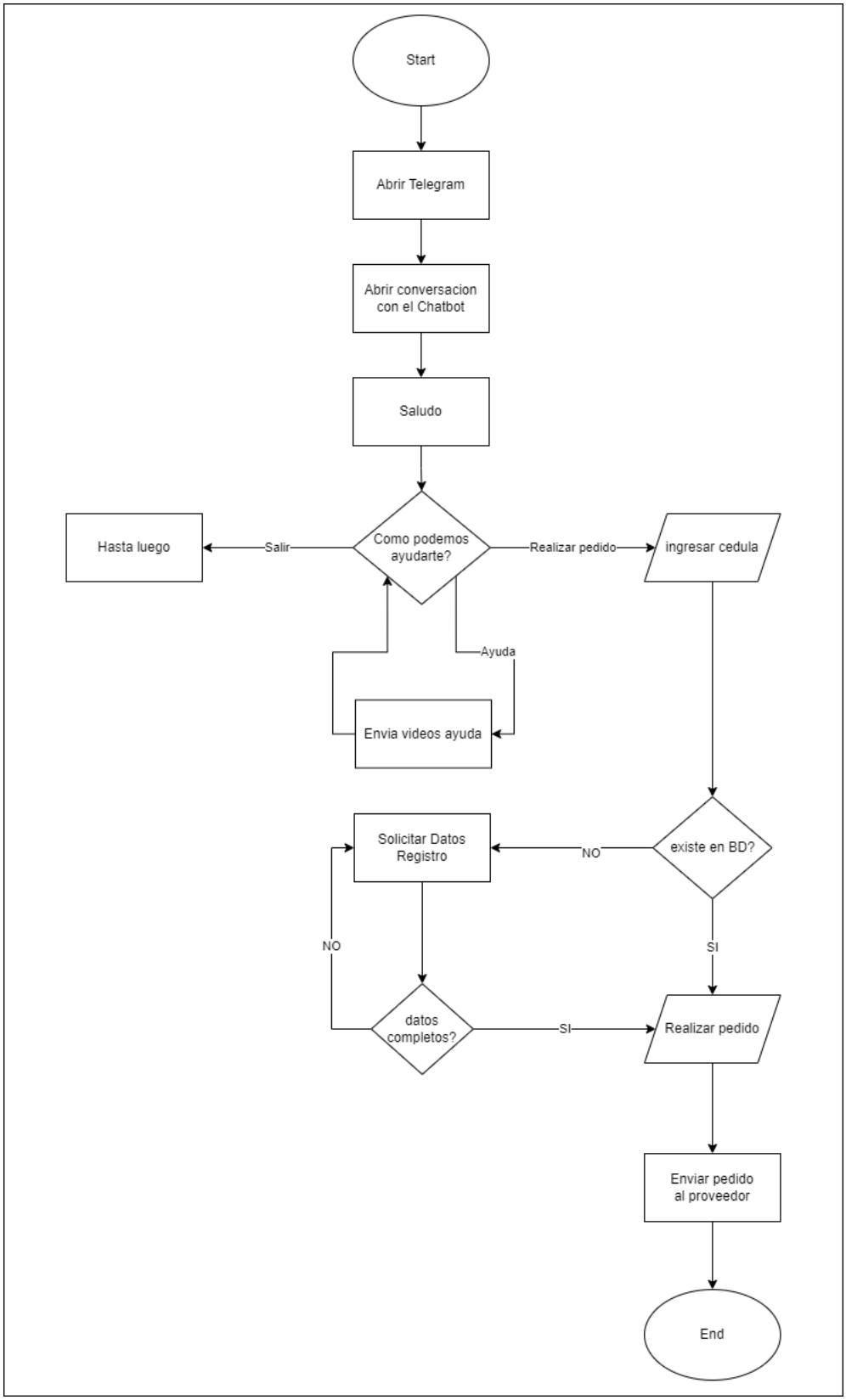


Figura 4.1: Diagrama de Flujo.

4.3. Arquitectura del Chatbot

4.3.1. Criterios de elección

Como punto de partida en el diseño de la arquitectura, se escogieron los siguientes atributos de calidad de acuerdo con los requisitos no-funcionales:

- Rendimiento
- Disponibilidad
- Mantenibilidad

Estos atributos cumplen con la mayoría de los requisitos no funcionales mencionados en la Figura 10.4 del capítulo de Anexos, que son RNF-2, RNF-3 y RNF-5, que se refieren a los tiempos de respuesta esperados para realizar ciertas funciones, la necesidad de una alta disponibilidad del software y la facilidad de mantenimiento. Teniendo en cuenta que estos atributos son esenciales para garantizar el correcto funcionamiento del Chatbot a desarrollar.

4.3.2. Arquitectura seleccionada

Teniendo en cuenta los criterios mencionados anteriormente, se llegó a la conclusión de que la arquitectura MVC (Modelo–vista–controlador) sería la elegida para realizar el proyecto del Chatbot debido a los grandes beneficios que esta puede llegar a aportar.

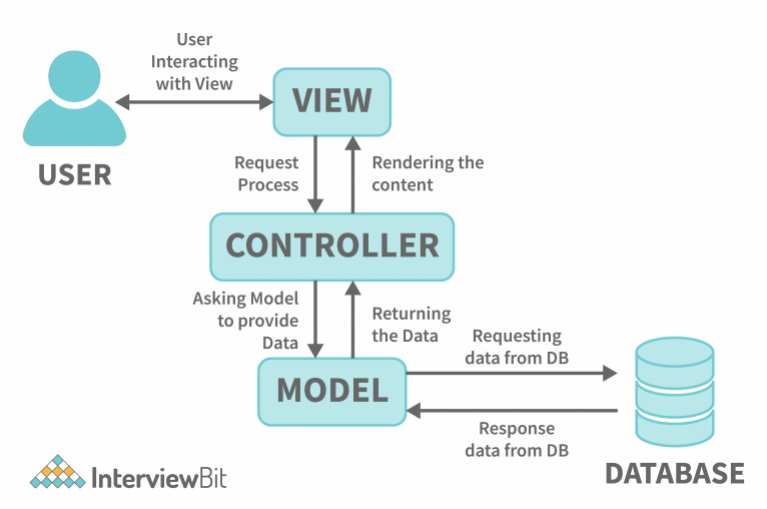


Figura 4.2: Diagrama patrón de diseño Modelo-Vista-Controlador [40].

En primer lugar, en cuanto a la disponibilidad, al poder tener los componentes de una manera independiente, se puede dar una solución rápida en el caso de que ocurra un fallo en alguno de los elementos de esta arquitectura.

En segundo lugar, como se puede apreciar en la Figura 4.2, la separación de responsabilidades entre las diferentes capas de la aplicación proporciona más modularidad y flexibilidad en el diseño. En particular, la capa del modelo maneja la lógica de negocios, la capa de la vista se encarga de la interfaz del usuario y la capa del controlador actúa como intermediario entre las otras dos capas.

Además, la arquitectura MVC permite una mayor reutilización del código, lo que reduce el tiempo y el esfuerzo necesarios para desarrollar la aplicación. También facilita el mantenimiento y evolución de la aplicación a lo largo del tiempo, ya que los cambios de la lógica de negocio se pueden realizar en la capa del modelo sin afectar a la vista o el controlador. Por último, la arquitectura MVC permite una mayor escalabilidad de la aplicación y una mejor distribución de la carga de trabajo entre diferentes componentes de la aplicación, lo que puede mejorar el rendimiento y la capacidad de respuesta del chatbot.

4.4. Modelo de datos

Para gestionar el flujo de información en este proyecto, se utilizó una base de datos relacional debido a su eficacia en el manejo de datos. Puesto que, un sistema de gestión de bases de datos relacional, como lo es PostgreSQL, es capaz de almacenar fácilmente cada tipo de dato y minimizar la presencia de datos vacíos. La Figura 4.3 muestra el modelo relacional de la base de datos del Chatbot.

Dentro de la base de datos relacional, las tablas principales son:

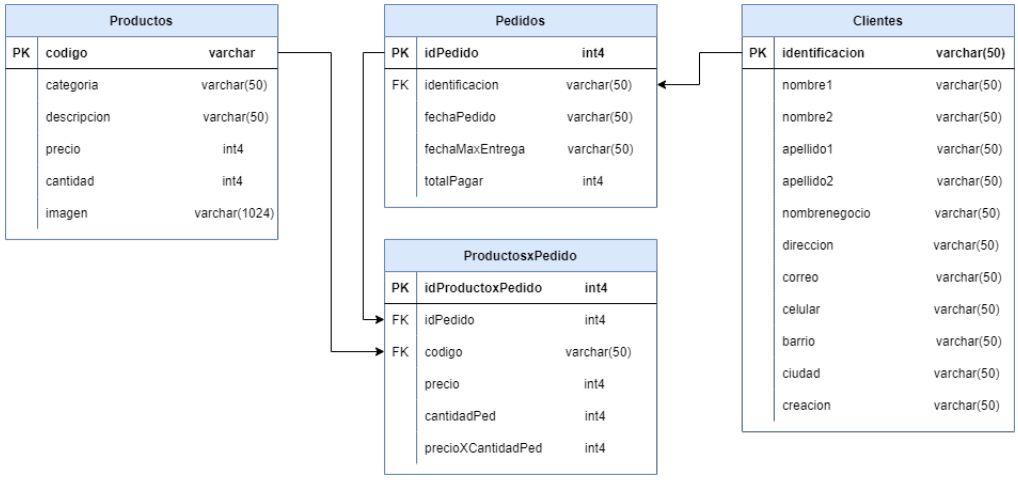


Figura 4.3: Modelo Relacional de Datos.

- Productos: almacena información relevante de los productos con los que cuenta un Proveedor en su inventario.
- Clientes: almacena la información personal y datos relevantes de los tenderos.
- Pedidos: almacena la identificación del cliente y los datos relevantes de un pedido.
- ProductosxPedido: cumple la función de una tabla intermedia que permite almacenar y asignar los distintos productos seleccionados previamente a un pedido por medio de su id.

4.5. Prototipo

Se utilizó la herramienta "Balsamiq Mockups"[41] para crear un prototipo del Chatbot, con el objetivo de reunir todas las características especificadas en los requisitos y obtener un modelo que se asemeje de manera precisa al Chatbot final que se pretende desarrollar.

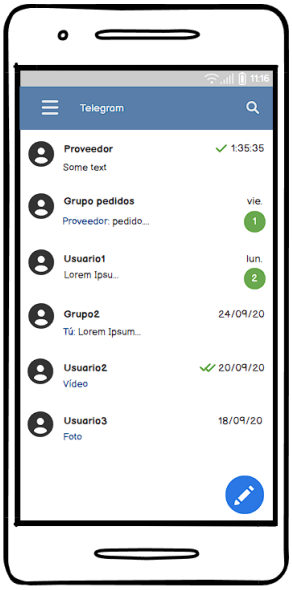


Figura 4.4: Vista pantalla principal.

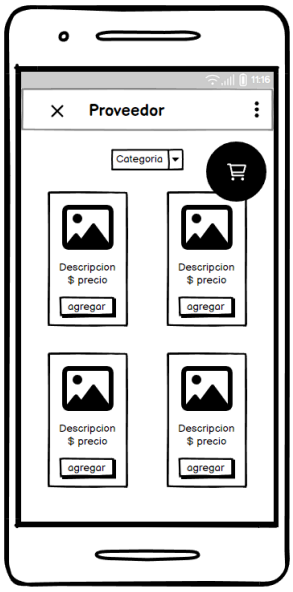


Figura 4.5: Vista selección categoría.

Es necesario revisar la sección 10.2 del Capítulo de Anexos para obtener más información sobre los prototipos o "mockups"que se han desarrollado durante la fase de diseño. Esta sección contiene documentación de todos los prototipos que se han desarrollado antes de empezar a trabajar en el software, y la sección 10.3, donde están las vistas finales que comprende el software.

4.5.1. Aprendizajes

Posteriormente a la realización de los prototipos planteados del Chatbot en Telegram, se decidió mostrar los mismos a los stakeholders con el fin de poder recibir una primera retroalimentación, la cual fue muy importante para poder continuar desarrollando el Chatbot de acuerdo con sus necesidades.

- Por la parte de los Tenderos se obtuvo la recomendación de incluir un botón de contacto que permita al usuario recibir los datos relevantes de su proveedor como lo son la dirección, el correo, teléfonos, redes sociales, etc. Con el fin de poder comunicarse con éste en caso tal de que sea requerido. Asimismo, se recomendó que, en la sección de ayuda, se cuente con alguna ayuda audiovisual (video) que explique el paso a paso de cómo se realizan tanto el proceso de registro, como el de realizar un pedido, lo cual facilitará su proceso de aprendizaje para poder interactuar con el Chatbot.

- Por la parte de los Proveedores se obtuvieron dos recomendaciones muy valiosas, la primera de ellas es la conexión de la base de datos con alguna herramienta similar a Excel, con el fin de que los Proveedores en un futuro puedan llegar a visualizar y manipular los distintos datos que se están almacenando, esto dado que la mayoría de ellos no cuentan con los conocimientos para acceder a una base de datos robusta a la cual le puedan realizar distintas consultas de acuerdo con su necesidad. La segunda recomendación, tiene más relación con la parte de la interfaz gráfica de la Web App del Chatbot con la que va a interactuar el Tendero, con su funcionalidad y con la experiencia del usuario, en la cual se propuso que al momento de realizar un pedido, debería de existir una pantalla de carga mientras se termina de completar el proceso del pedido, la cual mantenga informado al usuario de lo que está sucediendo, lo cual se alinea con los principios de usabilidad planteados anteriormente en el marco de referencia.

Capítulo 5

Tecnologías Involucradas en el Desarrollo del Chatbot

5.1. Introducción

En este capítulo se explican las razones tomadas con relación a la elección de las distintas tecnologías utilizadas para desarrollar el Chatbot. El capítulo se divide en tres secciones: frontend, backend y bases de datos. Estas elecciones se justifican teniendo en cuenta la naturaleza de la arquitectura y el enfoque del proyecto.

5.2. Frontend

Para el desarrollo de las vistas con las que va a interactuar el usuario dentro del Chatbot, se optó por una tecnología que fuera capaz de cumplir con los requisitos funcionales, no funcionales y de calidad especificados anteriormente, es por esto, que se seleccionó el framework de JavaScript creado por Facebook, ReactJS [42].

ReactJS es una opción recomendable, debido a su eficiencia al utilizar la herramienta del "Virtual DOM", lo que permite actualizar solo algunos componentes o partes de la página, en lugar de recargar toda la página, lo que lo hace muy rápido y eficiente en términos de rendimiento para aplicaciones de una sola página. Además, es muy flexible y se puede integrar fácilmente con otros frameworks y bibliotecas, lo que lo hace fácil de usar junto con otras tecnologías que pueda necesitar el proyecto. Otro beneficio de usar ReactJS es que como este utiliza componentes para dividir la interfaz de usuario en piezas pequeñas y reutilizables, permite simplificar el proceso de actualización y mantenimiento del código, logrando una mayor eficiencia en la creación de aplicaciones al construir un gran número de componentes reutilizables para diferentes páginas o secciones [43].

A diferencia de otros frameworks, se puede decir que ReactJS es bastante seguro debido a la manera que está diseñado, ya que este protege sus aplicaciones de los ataques XSS (cross-site scripting) que pueden llegar a realizar para intentar extraer datos de un usuario [44].

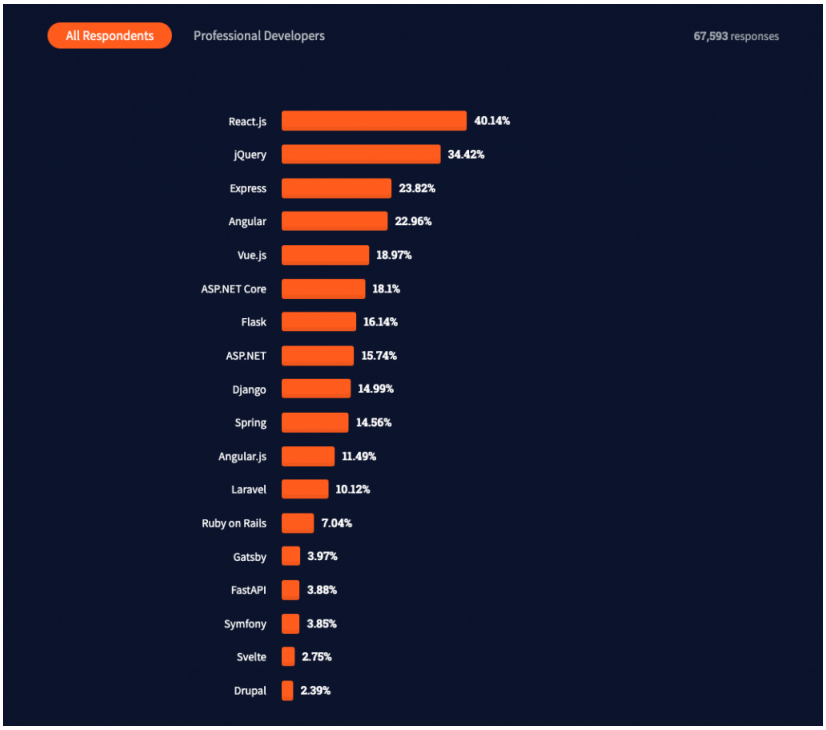


Figura 5.1: Popularidad del framework ReactJS en la comunidad desarrolladora [45].

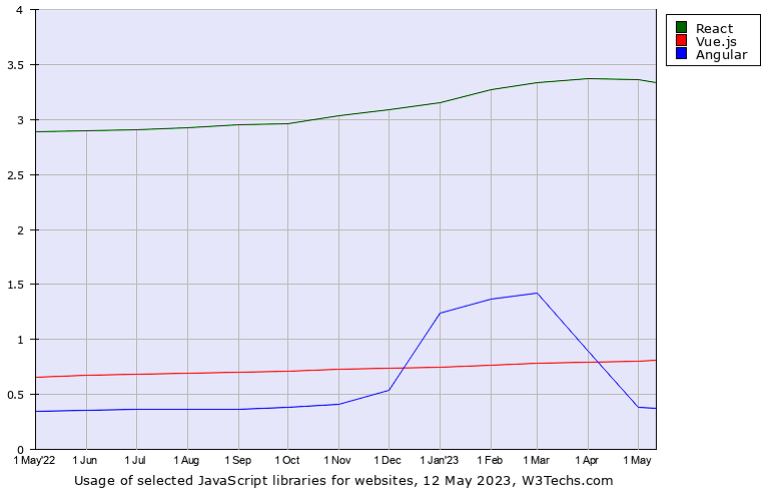


Figura 5.2: Popularidad del framework ReactJS para la creación de sitios web [46].

Por último, ReactJS tiene una gran comunidad de desarrolladores que trabajan constantemente en nuevas herramientas, bibliotecas y soluciones para diferentes problemas que pueden surgir en el desarrollo del proyecto. Como se puede apreciar en la Figura 5.1 y 5.2, el framework ha generado

un gran impacto en la comunidad de programadores debido a su curva de aprendizaje, así como su rendimiento excepcional, es por esto por lo que hoy en día aún se puede evidenciar la continua escogencia de esta tecnología para diferentes desarrollos de sitios web demostrando ser un framework bastante competitivo y popular en la comunidad desarrolladora [47].

5.3. Backend

Teniendo en cuenta los requisitos funcionales y no funcionales planteados, se decidió utilizar la plataforma de mensajería Telegram para la creación del Chatbot, esto debido a que Telegram, a diferencia de otras aplicaciones como WhatsApp, cuenta con una API bien documentada que facilita su integración con una aplicación web (WebApp) que permite a los usuarios interactuar directamente con el Chatbot, por lo que ofrece una amplia gama de características avanzadas y personalizables para los Chatbots, como por ejemplo, la creación de bots con múltiples idiomas, bots de encuestas, bots de juegos, etc. Asimismo, Telegram cuenta con un cifrado de extremo a extremo y medidas de seguridad adicionales que permiten evitar el uso malintencionado de los bots para proteger la privacidad de los usuarios [48].

Luego de indagar entre los distintos lenguajes de programación que tienen compatibilidad con la API de Telegram para la creación de un Chatbot [49], se optó finalmente por utilizar la librería pyTelegramBotAPI [50] de Python para la realización de este proyecto. Se escogió Python porque es un lenguaje de programación que cuenta con una sintaxis sencilla y fácil de leer, lo cual facilitará la creación y el mantenimiento de los bots que se puedan crear. Sumado a esto, Python cuenta con una amplia variedad de bibliotecas y módulos que permiten su integración con otros servicios, permitiendo su acceso a servicios web y bases de datos como por ejemplo SQL para almacenar información del bot que pueda resultar relevante. Por último, cabe destacar que Python, gracias a su versatilidad, cuenta con una gran comunidad activa de desarrolladores que comparten su conocimiento y brindan ayuda por medio de tutoriales o en distintos foros como Stack Overflow, por lo cual puede ser de gran ayuda por si en el desarrollo del proyecto surge algún inconveniente o duda [51], la Figura 5.3 muestra como Python impulsa a los diferentes desarrolladores a buscar Python después de tener un primer acercamiento básico en comparación con otros lenguajes de programación [52].

Figura 5.3: Preferencia de los diferentes lenguajes de programación en los últimos años [53].

En el momento de elegir un lenguaje de programación, es importante considerar también un framework que este pueda ofrecer, ya que estos proporcionan un entorno diseñado para simplificar el proceso de programación, puesto que incluyen diversos módulos y librerías con funciones ya implementadas, lo que permite ahorrar tiempo y esfuerzo al crear el código necesario. Además, el uso de un framework puede ayudar significativamente en la estructuración de la lógica de la aplicación y en la conexión con una base de datos.

Teniendo en cuenta lo anteriormente planteado y después de investigar minuciosamente sobre la manera en que se va a desarrollar el Chatbot, teniendo en cuenta los requisitos, la arquitectura MVC planteada y que este se comunicara con la API de Telegram por medio de un Webhook en el Backend, se decidió utilizar el framework de Flask, puesto que este es completamente compatible con el patrón MVC y permite realizar un buen manejo de las rutas por medio del controlador. Sumado a esto, se tiene que Flask es un framework minimalista que resulta ser altamente personalizable y flexible, ya que se puede integrar con distintas bibliotecas, extensiones y herramientas que permiten incrementar sus funciones, sin dejar de lado la rapidez y su buen desempeño [54]. Asimismo, Flask no requiere de una infraestructura con un servidor web para la realización de pruebas, dado que este cuenta con uno propio que permite ir observando los resultados que se van obteniendo, lo cual permite agilizar este proceso.

Por otro lado, se tiene que Flask cuenta con una curva de aprendizaje rápida dado que cuenta con una excelente documentación que es fácil de seguir y entender, lo cual ha permitido crear una gran comunidad de desarrolladores muy activa, lo que significa que existen muchos recursos y herramientas disponibles que permitirán ayudar en el proceso de desarrollo de la lógica del Chatbot.

5.4. Base de Datos

Con respecto al almacenamiento de los datos con los que trabajará el Chatbot, se optó por usar PostgreSQL, el cual es un sistema de gestión de bases de datos relacional (RDBMS). Esta herramienta manejará:

- Datos básicos del usuario Tendero que se registra o comunica con el Chatbot.
- Datos relevantes de los productos que ofrece un Proveedor.
- Datos necesarios del pedido solicitado por parte del Tendero al Proveedor.

Se decidió escoger una base de datos relacional debido a la sencillez que este modelo brinda para permitir manejar grandes cantidades de datos y soportar un alto rendimiento, ya que gracias a su estructura organizada, por medio del lenguaje SQL (Structured Query Language), se pueden realizar distintas consultas para relacionar o manipular los datos de las distintas tablas que se puedan llegar a tener, brindando también una gran flexibilidad y escalabilidad que permitirá asegurar que los datos manejados sean precisos y coherentes [55].

PostgreSQL es un sistema de gestión de bases de datos relacional (RDBMS) robusto, de código abierto y sofisticado que puede resultar beneficioso en el desarrollo de un Chatbot que requiere almacenar y acceder a información de manera eficiente, puesto que este ofrece características avanzadas como soporte para consultas complejas y transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) que garantizan que las transacciones de la Base de Datos se procesen de una manera fiable.

Asimismo, PostgreSQL es conocido por manejar grandes volúmenes de datos y altas cargas de trabajo, permitiendo el almacenamiento y consulta de varios tipos de datos, como texto, imágenes e incluso JSON, por lo que puede ofrecer una gran flexibilidad y un rendimiento sólido [56]. Por último, PostgreSQL, según las encuestas realizadas anualmente por Stack Overflow demuestra ser la primera opción a nivel de la comunidad desarrolladora, donde el 46 % de casi 49.000 programadores prefiere utilizar este gestor sobre otros existentes (Véase en la Figura 5.4).

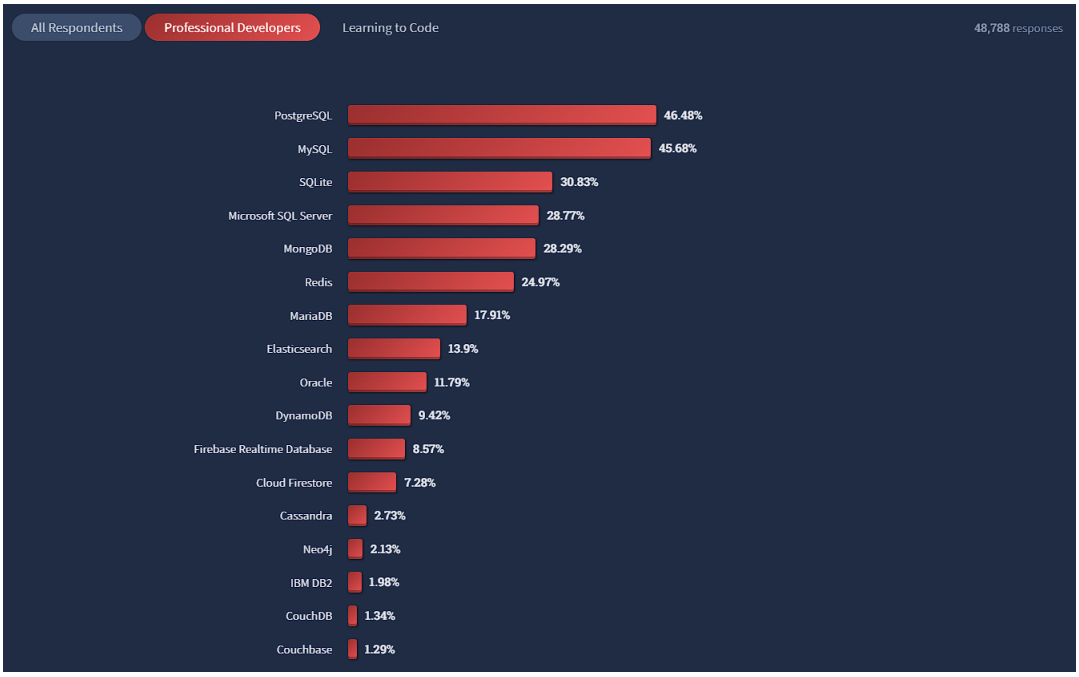


Figura 5.4: Top de los gestores de bases de datos preferidos en la comunidad desarrolladora [57].

Capítulo 6

Implementación del Software

6.1. Proceso de Desarrollo - Frontend

Debido al buen proceso de selección del framework de desarrollo para la parte "Frontend" del Chatbot propuesto, se podría decir que no hubo complicaciones al respecto. Dado que, tanto la instalación como la implementación del código en React fue sencilla y se adapta muy bien a los requisitos planteados anteriormente, ya que no se presentaron mayores dificultades en la creación de la interfaz del catálogo de productos del Proveedor y el carrito que permite ir almacenando lo que se va a pedir por parte del Tendero, esto gracias a que React cuenta con una amplia variedad de documentación, bibliotecas y herramientas disponibles; como por ejemplo los distintos "hooks" que permiten manipular o reutilizar componentes por medio de funciones, sin necesidad de usar clases.

A continuación, se mostrarán algunas secciones importantes del código en "React" que permitieron el correcto funcionamiento del Frontend del Chatbot:

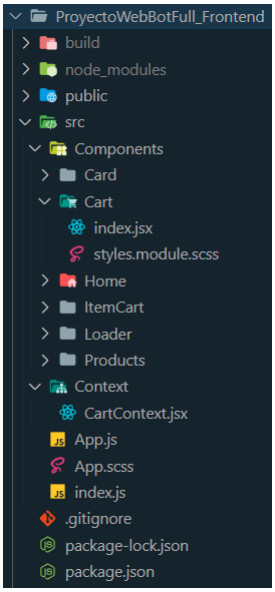


Figura 6.1: Implementación de componentes del Front en React.

Como se puede apreciar en la Figura 6.1, se crean distintos componentes estructurados haciendo

uso de una extensión de la sintaxis de JavaScript denominada JSX, la cual permite crear código con una sintaxis muy similar al HTML, mezclándolo con la lógica y demás funciones que ofrece JavaScript, Facilitando así el desarrollo de los componentes para la parte del Front. Asimismo, se puede apreciar que para darle estilos a cada uno de los componentes se hace uso de SCSS, el cual es una extensión o procesador CSS (Cascading Style Sheets), que permite generar hojas de estilo, añadiéndoles características que no tiene CSS, como pueden ser las variables, funciones, herencia, etc.

En los componentes creados se cuenta con el componente Home, el cual va a renderizar la página principal en la que se muestran los distintos productos con los que cuenta el proveedor, haciendo uso del componente Card, el cual es una carta que muestra una imagen del producto junto con su descripción, precio y un botón para añadir el producto al carrito. Este funciona en conjunto con la función Products que es la que permite listar y filtrar las distintas categorías de las cartas que se van a mostrar al Tendero en la página. Asimismo, se cuenta con un componente Cart que hace referencia a un carrito de compras en el cual se van a ir agregando distintos productos como se mencionaba anteriormente que se podrán visualizar dentro del carrito mismo gracias al componente ItemCart que es el que contiene la información del producto añadido, la cantidad de este y permitirá añadir o quitar un producto del carrito, para finalmente poder realizar el pedido por medio de un botón. Finalmente, se cuenta con un componente Loader que, como su nombre lo indica, está destinado para que una vez se realice el pedido, mientras se envían los datos necesarios al back, este sea renderizado para mantener informado al usuario de que se está procesando dicho pedido.

Figura 6.2: Contexto del Front.

En la Figura 6.2, se puede apreciar que se crea un elemento contexto haciendo uso de createContext, el cual permite establecer una comunicación para compartir datos y funciones entre los distintos componentes que accedan a él, esto se da gracias al provider, que es el que permite que los componentes que lo consumen se suscriban a los cambios del contexto. Permitiendo pasar datos a través del árbol de componentes sin tener que pasar propiedades manualmente en cada nivel. Asimismo, se puede apreciar que dentro del contexto se hacen uso de distintos "Hooks" de React, como useEffect y useState, que permiten trabajar con el estado, asignación de variables y los efectos que se le pueden dar a los componentes de React, como se puede apreciar en la línea 32 en la cual se hace un llamado a la función asíncrona getProducts, que es la que permite traer los distintos productos del proveedor, haciendo uso de la librería Axios para realizar una solicitud HTTP al endpoint del Backend.

Dentro del contexto también se establecen otras funciones importantes que no se llegan a visualizar en la imagen, como la de añadir un producto al carrito de compras, eliminar un producto del carrito de compras y la de realizar el pedido, las cuales son retornadas al final para poder compartirse con los demás elementos por medio del context y el provider.

ProyectoWebBotFull_Frontend > src > App.js > ...

```
1  import Home from "./Components/Home";
2  import {CartProvider} from "./Context/CartContext";
3  import { BrowserRouter as Router, Route, Routes } from 'react-router-dom'
4  import "./App.scss";
5
6  const App = () => {
7    /* Envolvemos la home con el provider del context */
8    return (
9      <CartProvider>
10       <Router>
11         <Routes>
12           <Route exact path='/' element={<Home/>} />
13         </Routes>
14       </Router>
15     </CartProvider>
16   );
17 };
18
19 export default App;
```

Figura 6.3: Definición de rutas Front.

ProyectoWebBotFull_Frontend > src > index.js > ...

```
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import App from './App';
4
5
6  const root = ReactDOM.createRoot(document.getElementById('root'));
7  root.render(
8
9      <App />
10
11 );
```

Figura 6.4: Renderizado Front.

Como se puede apreciar en las Figuras 6.3 y 6.4, se hace uso de la librería react-dom y react-router-dom para inicialmente en el archivo App.js declarar la ruta de navegación que se va a utilizar, en este caso para poder acceder al elemento home, envolviéndolo con la información del contexto por medio del provider para que tenga acceso a la misma, exportando finalmente el elemento para que posteriormente sea renderizado junto con su ruta especificada.

6.2. Proceso de Desarrollo - Backend

Debido al buen proceso de selección del lenguaje de programación Python junto con su framework Flask, a la experiencia y al buen manejo que se posee para desarrollar con este lenguaje, se puede decir que no se presentaron mayores inconvenientes en el proceso de desarrollo de la lógica del Chatbot. Sin embargo, si se tuvo problemas al inicio en la conexión del Chatbot con la lógica del Backend, ya que en un principio se quería hacer uso de una plataforma de comprensión de lenguaje natural de Google muy potente denominada Dialogflow [58], la cual permite diseñar la manera en la que se va a comunicar el usuario con el chatbot y es fácil de integrar con plataformas de mensajería instantánea como Telegram, pero en el momento de realizar la conexión por medio de un webhook, se observó que Dialogflow cuenta con algunas limitaciones como por ejemplo que no se puede acceder al chatId de la conversación de telegram por lo cual no se le pueden enviar ciertos mensajes o identificar específicamente con quién está hablando el Chatbot. Es por esto, que se decidió crear el Chatbot de manera manual haciendo uso de la librería pyTelegramBotAPI [50], para poder desarrollar el Chatbot de acuerdo con los requisitos y sin ninguna restricción.

A continuación, se mostrarán algunas secciones importantes del código en Python, que junto con el framework Flask y la librería pyTelegramBotAPI, permitieron realizar de manera correcta el "Backend" del Chatbot:

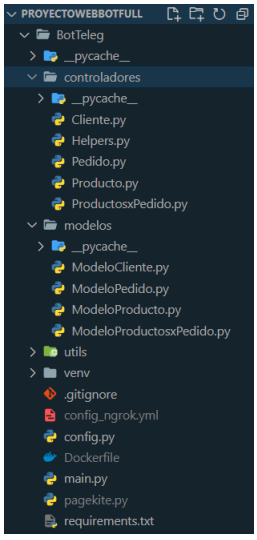


Figura 6.5: Estructura de archivos MVC.

Como se puede apreciar en la Figura 6.5, se estructuran o acomodan los distintos archivos de acuerdo con la arquitectura o patrón de diseño MVC (modelo-vista-controlador), estipulado anteriormente en el capítulo 4 de Diseño del Software. Organizando los distintos archivos dentro de las carpetas que hacen referencia a los componentes del modelo y el controlador, ya que la vista en este caso, que es la interfaz de la Web App que se va a mostrar dentro de la conversación con el Chatbot, se encuentra aparte en la parte del Frontend desarrollado en React y mencionado anteriormente. Asimismo, se puede ver que cada uno de los modelos de las tablas creadas en la base de datos, cuenta con un controlador, el cual cuenta con funciones CRUD (Create, Read, Update, Delete) que pueden ser utilizadas más adelante para manipular los datos de las distintas tablas.

BotTeleg > main.py > ...

```
59  #Gestiona las peticiones o actualizaciones POST enviadas al webhook
60  #del bot en el servidor web y las procesa:
61  @web_server.route('/', methods=['POST'])
62  def webhook():
63      #si el post recibido es un JSON:
64      if request.headers.get("content-type") == "application/json":
65          update = telebot.types.Update.de_json(request.stream.read().decode('utf-8'))
66          bot.process_new_updates([update])
67          return "OK",200
68
69
70  #responde al comando /start :
71  @bot.message_handler(commands=['start'])
72  def cmd_start(message):
73      """ Muestra las acciones disponibles al iniciar una conversacion. """
74
75      bot.send_message(message.chat.id,"Buen dia, se ha comunicado con Distri Romel Sas")
76
77      #definimos tres botones
78      markup= ReplyKeyboardMarkup(one_time_keyboard=True,
79                                  input_field_placeholder="Pulsa un boton",
80                                  resize_keyboard=True)
81      markup.add("Realizar pedido","Contactenos","Ayuda","Salir")
82      #preguntamos por la accion
83      msg = bot.send_message(message.chat.id, "Como podemos ayudarle?",reply_markup=markup)
84      bot.register_next_step_handler(msg,realizar_pedido)
```

Figura 6.6: Definición del Webhook.

En la Figura 6.6, se puede ver la función webhook creada, la cual resulta ser fundamental para establecer la comunicación del Chatbot con la Api de Telegram, esta función permite ir recibiendo todo tipo de peticiones o actualizaciones que se realizan al Chatbot durante una conversación, por medio del método “POST”, para poder después procesar dicha información utilizando la librería pyTelegramBotAPI, para darle una respuesta acertada al usuario que permita continuar con el flujo de la conversación y la lógica establecida. Como se puede percibir en la función cmd_start, en la línea 72, la cual es una función que responde al comando “/start” que inicia una conversación por primera vez con un bot de Telegram.

Figura 6.7: Web App dentro del Chatbot.

Otra sección fundamental del código se puede apreciar en la Figura 6.7, ya que dentro de esta función corroborar_cedula, se verifica si un usuario (tendero) ya se encuentra registrado en la base de datos del proveedor, para posteriormente en caso tal de que así sea, se le envía un mensaje con un botón el cual es el que permite abrir la Web App especificada en la línea 216 mediante su URL para poder tomar su pedido. Por otro lado, en caso tal de que el usuario no exista en la base de datos, se almacena en un diccionario su cédula y se empezará el proceso de registro correspondiente, preguntando y almacenando en el mismo sus datos requeridos, para poder finalmente completar su registro y almacenarlo en la base de datos.

```
BotTeleg > main.py > recibePedido
411  #almacena datos del usuario y del bot para
412  #que despues estos puedan compararse con
413  #la informacion solicitada por el front:
414  @web_server.route('/botdata',methods=['GET','POST'])
415  async def datosbot():
416      if request.method == 'POST':
417          data = request.get_json() # se obtienen los datos del request que llegan al endpoint
418          chatid = data["chatId"]
419          identificacion = data["identificacion"]
420
421
422          usuariosAct[chatid]={}
423          usuariosAct[chatid]["identificacion"] = identificacion
424          usuariosAct[chatid]["chatid"] = chatid
425
426
427          print("llamado del back:")
428          print("Data:",data)
429          print("chatid:",chatid)
430          print("identificacion:",identificacion)
431
432          return data
433
434
435  #Endpoint que recibe el pedido que llega desde el front:
436  @web_server.route('/recibePedido',methods=['GET', 'POST'])
437  async def recibePedido():
438
439      if request.method == 'POST':
440          print("WEEEEEE")
441          req = request.get_json(silent=True, force=True) #ensayar con esto
442          res = json.dumps(req, indent=4)
443          print(res)
444          ped = json.loads(res)
445          pedidoCliente = ped
446          print("pedido:\n",pedidoCliente)
```

Figura 6.8: Endpoints importantes.

Por último, en la Figura 6.8 se muestran los endpoints datosbot y recibePedido, los cuales gracias al comportamiento de Flask se pueden declarar asíncronos, permitiendo crear un hilo de ejecución que ejecuta una corrutina o función aparte, por cada conversación que se está realizando con el Chatbot. Esto permite que, por ejemplo, con la función datosbot se pueda ir almacenando en un diccionario las "sesiones" por medio del chatId que es diferente para cada conversación en Telegram, junto con su identificación o cédula de los usuarios que están interactuando, lo que permite distinguir al Chatbot con qué usuario está tratando en todo momento. Asimismo, la función recibePedido como su nombre lo indica, es la encargada de recibir de la parte del Front un JSON que contiene el pedido con los productos seleccionados por el usuario, junto con su identificación correspondiente, que posteriormente serán datos que serán procesados, almacenados en sus respectivas tablas y enviados al Proveedor para completar la cadena de suministro.

6.3. Proceso de Desarrollo - Base de Datos

Para la implementación de la base de datos del Chatbot, se modeló primero el modelo relacional que se puede apreciar anteriormente en la Figura 4.3 del capítulo de diseño del software. Este se realizó haciendo uso del software "draw.io" [59], el cual es un software gratuito y de código abierto desarrollado en HTML5 y Javascript, el cual permite crear distintos tipos de diagramas que pueden ser útiles en el proceso del desarrollo del software y que en este caso fue fundamental para modelar la base de datos creada en PostgreSQL. Para la creación de la base de datos, se puede decir que no hubo inconvenientes gracias a la experiencia previa que se tenía con el motor de base de datos seleccionado.

A continuación, se mostrarán algunas secciones importantes, que permitieron la creación y correcto funcionamiento de la base de datos.

BotTeleg > modelos > ModeloProducto.py > ...

```python
from config import db

You, hace 4 meses | 1 author (You)
class Producto(db.Model):

    __tablename__="productos"

    codigo=db.Column(db.String, primary_key=True)
    categoria=db.Column(db.String)
    descripcion=db.Column(db.String)
    precio=db.Column(db.Integer)
    cantidad=db.Column(db.Integer)
    imagen= db.Column(db.String)

    def __repr__(self):
        return f"Producto: {self.codigo,self.categoria,self.descripcion,self.precio,self.cantidad,self.imagen}"

    def __init__(self,codigo,categoria,descripcion,precio,cantidad,imagen):
        self.codigo = codigo
        self.categoria = categoria
        self.descripcion = descripcion
        self.precio = precio
        self.cantidad = cantidad
        self.imagen = imagen

def format_producto(producto):
    return{
        "codigo":producto.codigo,
        "categoria":producto.categoria,
        "descripcion":producto.descripcion,
        "precio":producto.precio,
        "cantidad":producto.cantidad,
        "imagen":producto.imagen
    }
```

Figura 6.9: Modelo de tablas BD.

Como se puede apreciar en la Figura 6.9, se utilizaron los modelos establecidos en el código del Backend para la creación de las distintas tablas que se van a utilizar. En este ejemplo se puede ver cómo se crea la clase Producto haciendo uso del kit de herramientas SQL y ORM (Object Relational Mapper) para Python "SQLAlchemy" [60], la cual corresponde a la tabla productos creada posteriormente y que cuenta con atributos como codigo que se define como la primary key, categoria, descripcion, precio, cantidad e imagen.

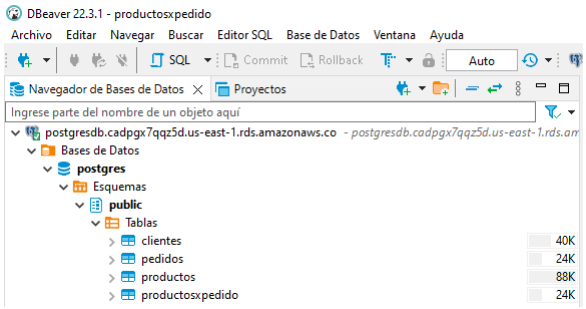


Figura 6.10: Tablas Creadas Base de Datos.

En la Figura 6.10 se puede evidenciar la creación de las 4 tablas clientes, productos, pedidos y productosxpedido en la base de datos postgres, la cual se encuentra alojada en un servicio de bases de datos relacionales de Amazon (Amazon RDS), el cual se ejecuta "en la nube" y está diseñado para simplificar el funcionamiento, configuración y escalabilidad de una base de datos relacional para su uso en distintas aplicaciones [61].

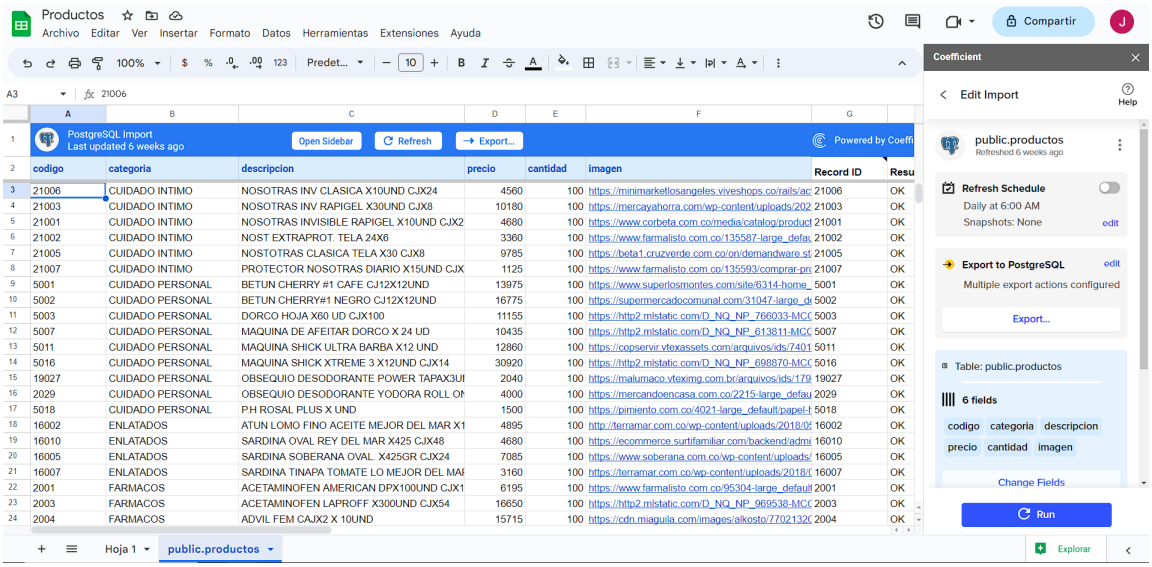


Figura 6.11: Conexión Base de Datos y Excel.

Teniendo en cuenta el requisito funcional con identificación "RF-29", planteado en la sección 10.1.1 del capítulo 10.1 de Requisitos, en el cual dice que "El chatbot debe poder integrar las bases de datos con Excel para poder facilitar al proveedor modificar el inventario y así mismo, poder importar y exportar datos." Se busca la manera de conectar la base de datos junto con un Excel en Google Drive utilizando una extensión llamada Coefficient [62], la cual es una herramienta que permite conectar las hojas de cálculo de Google con las bases de datos de Salesforce, MySQL y PostgreSQL. Permitiendo actualizar, modificar, importar y exportar datos, e incluso poder programar

actualizaciones o alertas que permitan monitorear las distintas tablas. Véase la Figura 6.11.

6.4. Proceso de Desarrollo - Despliegue

En el diagrama a continuación se quiso representar las distintas herramientas que se utilizaron para lograr programar el Chatbot y conectarlo a la API de Telegram correctamente, teniendo en cuenta la arquitectura MVC planteada, los distintos requisitos planteados y la interacción del usuario. Véase la Figura 6.12.

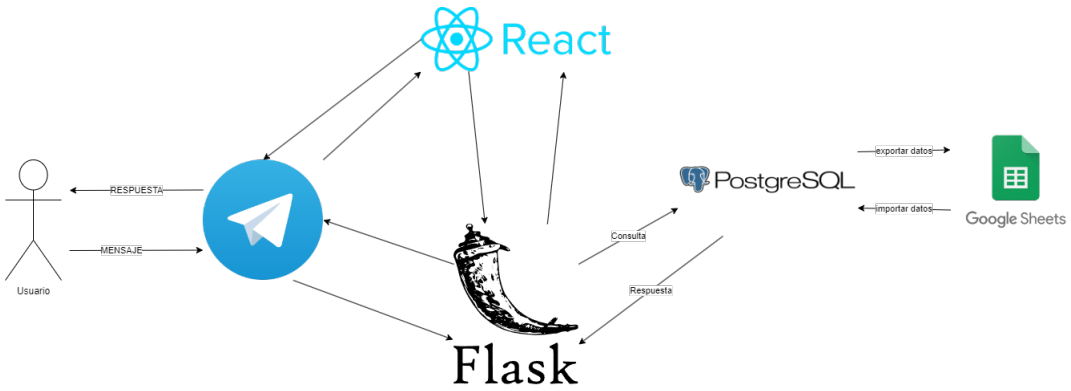


Figura 6.12: Diagrama Herramientas Desarrollo.

En el siguiente Diagrama se muestran las distintas plataformas o herramientas que se utilizaron posteriormente para poder realizar el despliegue del código del Chatbot. Véase la Figura 6.13.

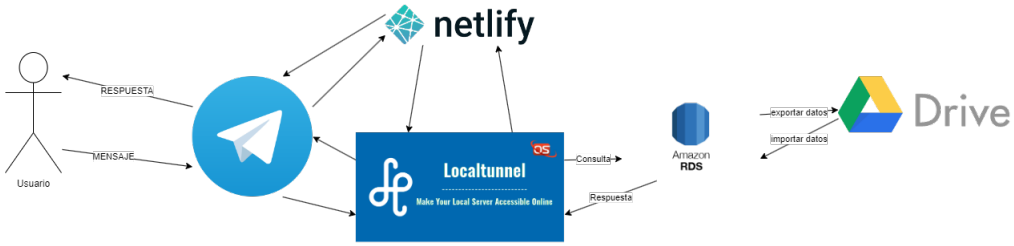


Figura 6.13: Diagrama Herramientas Despliegue.

Como se puede apreciar en el diagrama, para realizar el despliegue de la parte del "Frontend", se utilizó la plataforma de Netlify [63], la cual permite alojar y mantener un sitio web estático o aplicación con implementación continua y protocolo HTTPS. Asimismo, para la parte del "Backend", se hizo uso de la herramienta LocalTunnel [64], la cual permite exponer un servidor web local como el que se tiene en Flask a través de internet de manera temporal, se escogió esta herramienta porque a diferencia de otras plataformas como Ngrok [65], LocalTunnel permite crear subdominios de manera gratuita con una URL con certificado HTTPS, única, estática y pública

que se asocia con el servidor local que se quiere exponer. Lo cual resulta conveniente para desplegar y conectar este prototipo del Chatbot de manera correcta, tanto con Netlify como con Telegram que solicita ciertos protocolos para poder conectarse a la API. Por último, se utilizó Google Drive para la creación de las hojas de cálculo que se van a comunicar con la base de datos PostgreSQL mencionada anteriormente en el proceso de desarrollo de la base de datos y que se desplegó en un RDS de AWS (Amazon Web Services).

6.5. Prototipo Final Funcional

A continuación, se mostrarán algunas de las vistas relevantes obtenidas del prototipo final del Chatbot funcionando en Telegram.

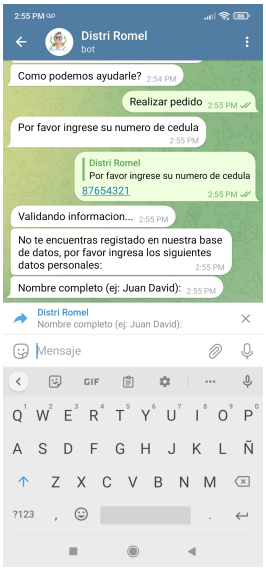


Figura 6.14: Vista proceso de Registro Tendero Parte III.

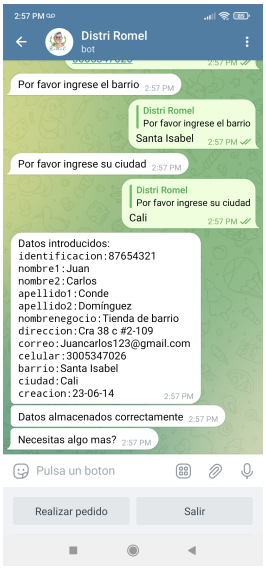


Figura 6.15: Vista proceso de Registro Tendero Parte IV.

Figura 6.16: Vista proceso Realizar pedido Parte IV.

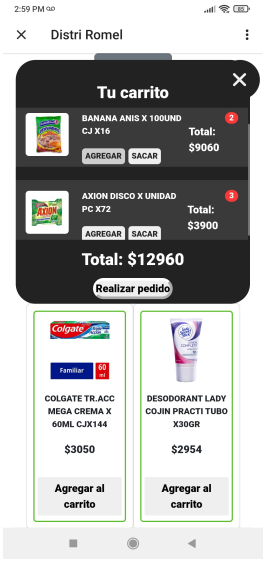


Figura 6.17: Vista proceso Realizar pedido Parte V.

Para ver detalladamente cada una de las vistas es necesario revisar la sección 10.3 del Capítulo de anexos, donde están las vistas finales que comprende el Prototipo Final. Asimismo, si se desea ver el funcionamiento del Chatbot, se recomienda ver el siguiente video disponible en: https://youtu.be/0L8cFLV-IIg

Capítulo 7

Pruebas

7.1. Pruebas Funcionales

La sección presente contiene algunas pruebas funcionales realizadas para verificar el correcto funcionamiento e implementación del Chatbot desarrollado teniendo en cuenta los requerimientos planteados anteriormente. Para ello se efectuaron algunos casos de prueba que tenían en cuenta diferentes condiciones o variables, que se utilizan para determinar si una aplicación o software funciona de manera aceptable o no.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Iniciar una conversación con el chatbot en Telegram.	
Identificador	CP_FUN_001	
Nombre del Caso	Prueba de bienvenida	
Precondiciones	• Encontrarse dentro de la aplicación de Telegram	
Paso	**Resultado esperado**	**Resultado real**
1) Ir a la opcion "buscar" y buscar el chatbot nombrado "Distri Romel"	El chatbot debe enviar un mensaje de bienvenida y proporcionar opciones o instrucciones claras para el usuario.	**OK**
2) Ingresar en la conversacion y presionar la opcion "iniciar"		

Figura 7.1: Caso de prueba CP_FUN_001.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Validar Registro Usuario Tendero	
Identificador	CP_REG_001	
Nombre del Caso	Prueba de registro usuario tendero	
Precondiciones	• Encontrarse dentro de la conversacion con el Chatbot • No haberse registrado previamente.	
Paso	**Resultado esperado**	**Resultado real**
1) Presionar el boton de "Realizar Pedido"	El chatbot debe procesar los datos ingresados por el usuario y enviar una respuesta que indique que se ha registrado exitosamente	**OK**
2) Ingresar los datos solicitados por el Chatbot		

Figura 7.2: Caso de prueba CP_REG_001.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Validar Proceso Realizar Pedido	
Identificador	CP_PED_001	
Nombre del Caso	Prueba de validacion usuario tendero	
Precondiciones	• Encontrarse dentro de la conversacion con el Chatbot • Haberse registrado previamente.	
Paso	**Resultado esperado**	**Resultado real**
1) Presionar el boton de "Realizar Pedido"	El chatbot debe validar que el usuario se encuentra registrado en la BD y habilitar o responder con un boton que va a permitir abrir la Web App para realizar el pedido	**OK**
2) Ingresar la identificacion solicitada por el Chatbot		

Figura 7.3: Caso de prueba CP_PED_001.

Para ver detalladamente cada uno de los casos de prueba, es necesario revisar la sección 10.6 del Capítulo de anexos, donde están todos los casos de prueba que comprende el Prototipo Final. Asimismo, si se desea ver el funcionamiento completo del Chatbot, se recomienda ver el siguiente video disponible en: `https://youtu.be/0L8cFLV-IIg`

7.2. Pruebas de Usabilidad

Para esta parte de las pruebas se decidió llevar a cabo una encuesta, puesto que las encuestas son una herramienta para la investigación de mercados, que permite obtener información relevante de las personas encuestadas mediante el uso de cuestionarios previamente elaborados para obtener una información más específica [66]. La encuesta se realizó de manera personal, lo cual, permite recolectar información sobre distintos aspectos sociales, personales y económicos de las personas que forman parte de la investigación; con el fin de obtener datos de interés y sus opiniones.

La encuesta anteriormente mencionada se realizó para poder conocer la valoración u opinión de los usuarios en relación con los aspectos de usabilidad y adaptabilidad del software creado. Es por esto por lo que el cuestionario utilizado en la encuesta tenía como objetivo principalmente determinar en qué medida el software cumplía con los 10 principios de usabilidad de Jakob Nielsen, planteados anteriormente en el marco de referencia, lo cual explica por qué la mayoría de las preguntas planteadas se encuentran relacionadas con este aspecto.

La población de estudio seleccionada para la realización de las encuestas fueron los distintos administradores o encargados de las tiendas de barrio (Tenderos) y Proveedores que le surten a las mismas en la ciudad de Cali, Colombia.

7.2.1. Resultados Usuario Tendero

En esta sección, se solicita al encuestado que indique cuál es su edad y que tipo de plan de datos utiliza. Siendo estos, datos importantes que permiten analizar inicialmente el rango de edades de las personas que administran las tiendas de barrio, obteniendo un promedio de 46 años, lo cual es muy similar a los datos expuestos anteriormente en el planteamiento del problema por FENALCO. Asimismo, como se puede evidenciar en la Figura 7.11, nos podemos dar cuenta que la mayoría de los tenderos encuestados utilizan un plan de datos prepago en su celular.

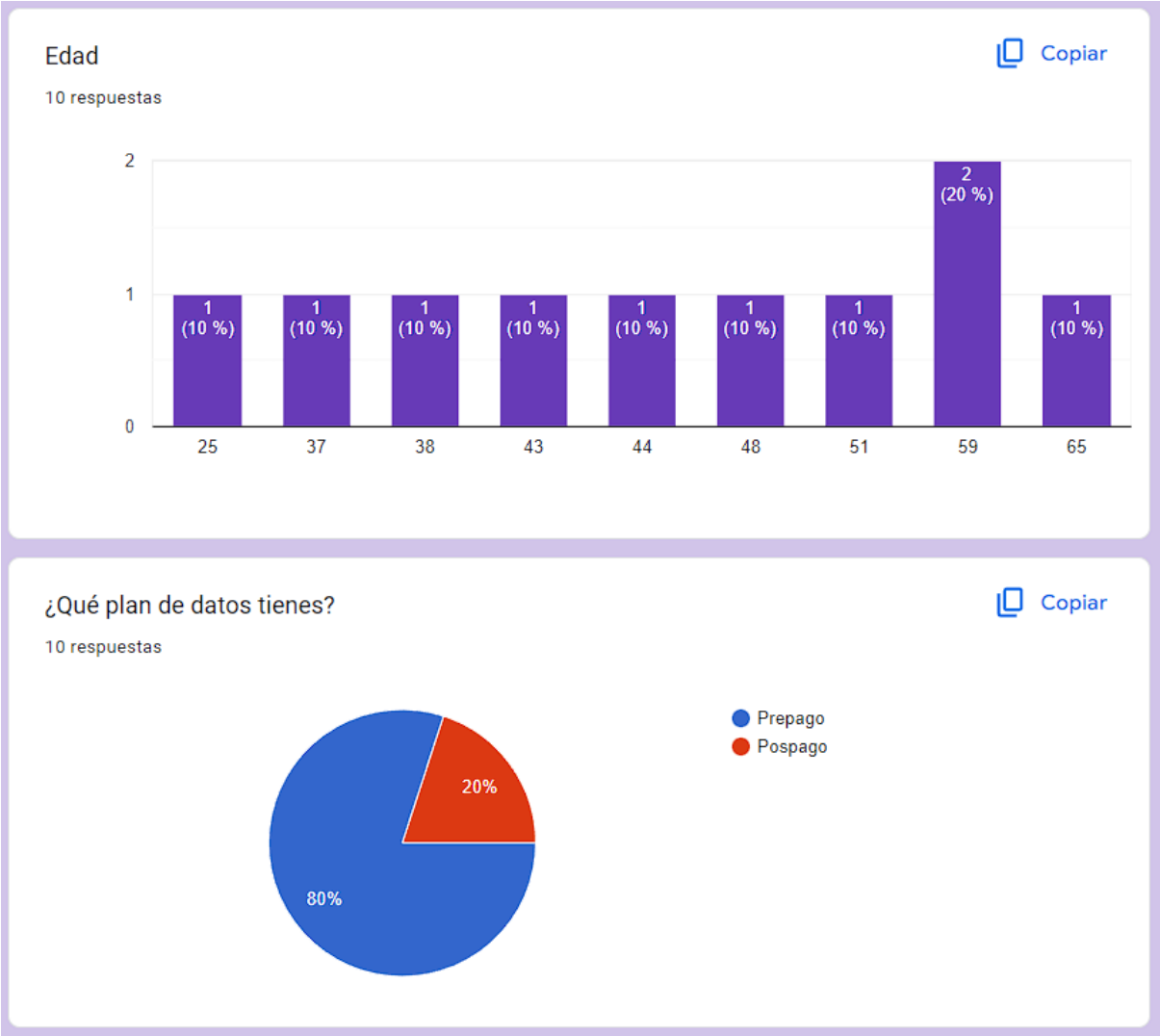


Figura 7.4: Encuesta Usuario Tendero Parte I.

En esta sección, el encuestado responde a las preguntas número 1 y 2, las cuales hacen referencia a los principios número 1 y número 2 de usabilidad, que indican que el usuario siempre debe mantenerse informado de su estado o de lo que está sucediendo, y, que se debe utilizar un lenguaje con palabras y expresiones familiares para el usuario. Evidenciando en la Figura siguiente, que el 100 % de los encuestados opinaron que el Chatbot cumplía totalmente con ambos principios de usabilidad.

Figura 7.5: Encuesta Usuario Tendero Parte II.

En esta sección, el encuestado responde a las preguntas números 3 y 4, las cuales hacen referencia a que un usuario debe poder deshacer o repetir una acción previamente realizada, y, que el Chatbot debe mantener ciertas convenciones lógicas en cuanto a su presentación y diseño. Evidenciando en la Figura siguiente, que el 100 % de los encuestados opinaron que el Chatbot cumplía con el principio número 4 descrito anteriormente. Pero en cuanto al principio número 3, aunque se cumple dicho principio en un 90 %, hubo una persona que quizás no noto esta funcionalidad.

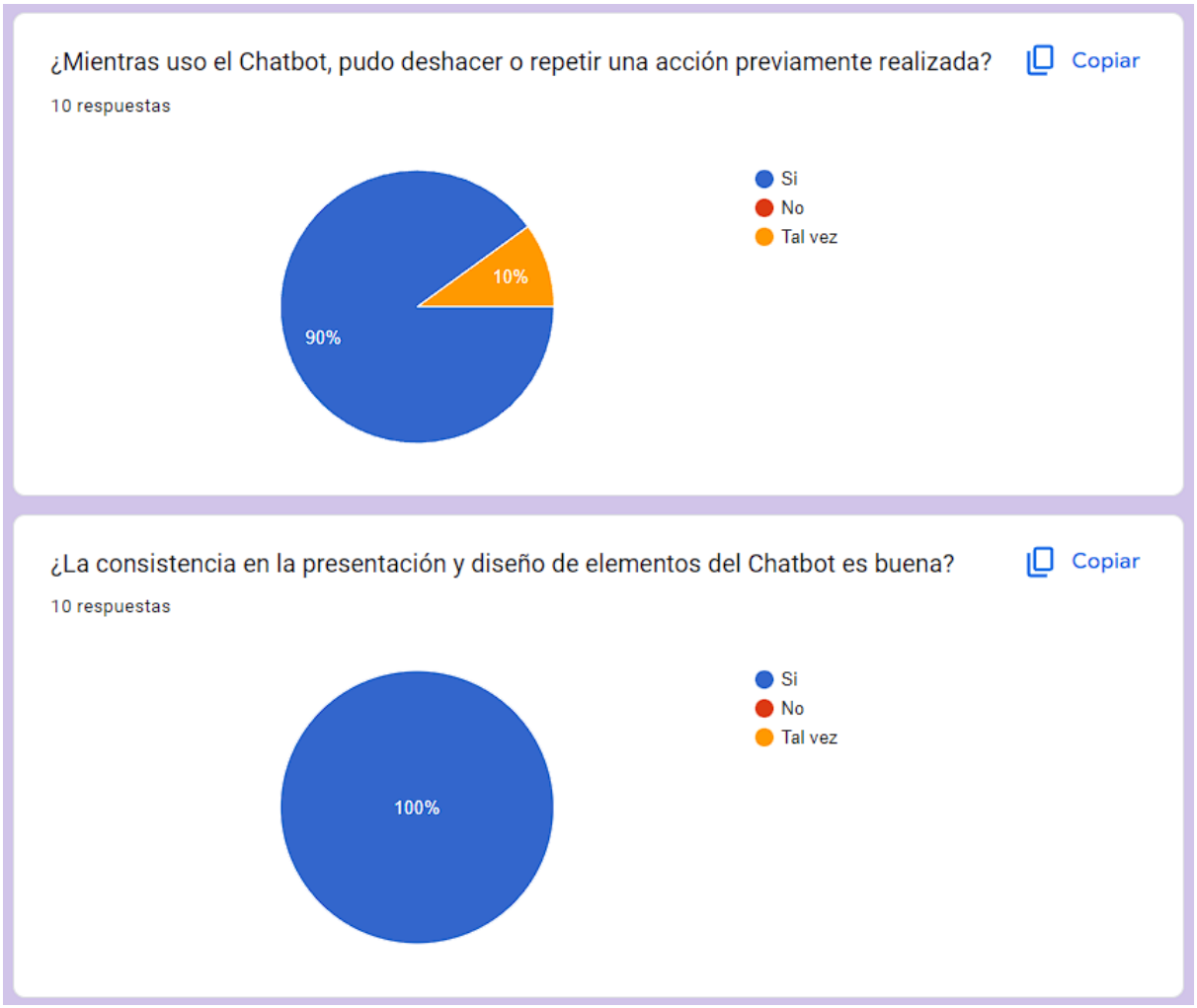


Figura 7.6: Encuesta Usuario Tendero Parte III.

En esta sección, el encuestado responde a las preguntas números 5 y 6, las cuales hacen referencia a que el Chatbot debe estar preparado para ayudar al usuario en caso tal de que éste cometa un error. Asimismo, que las opciones y acciones del Chatbot deben ser intuitivas para que el usuario no tenga que memorizar información mientras hace uso de este. Evidenciando en la Figura siguiente, que el 100 % de los encuestados opinaron que el Chatbot cumplía totalmente con ambos principios de usabilidad.

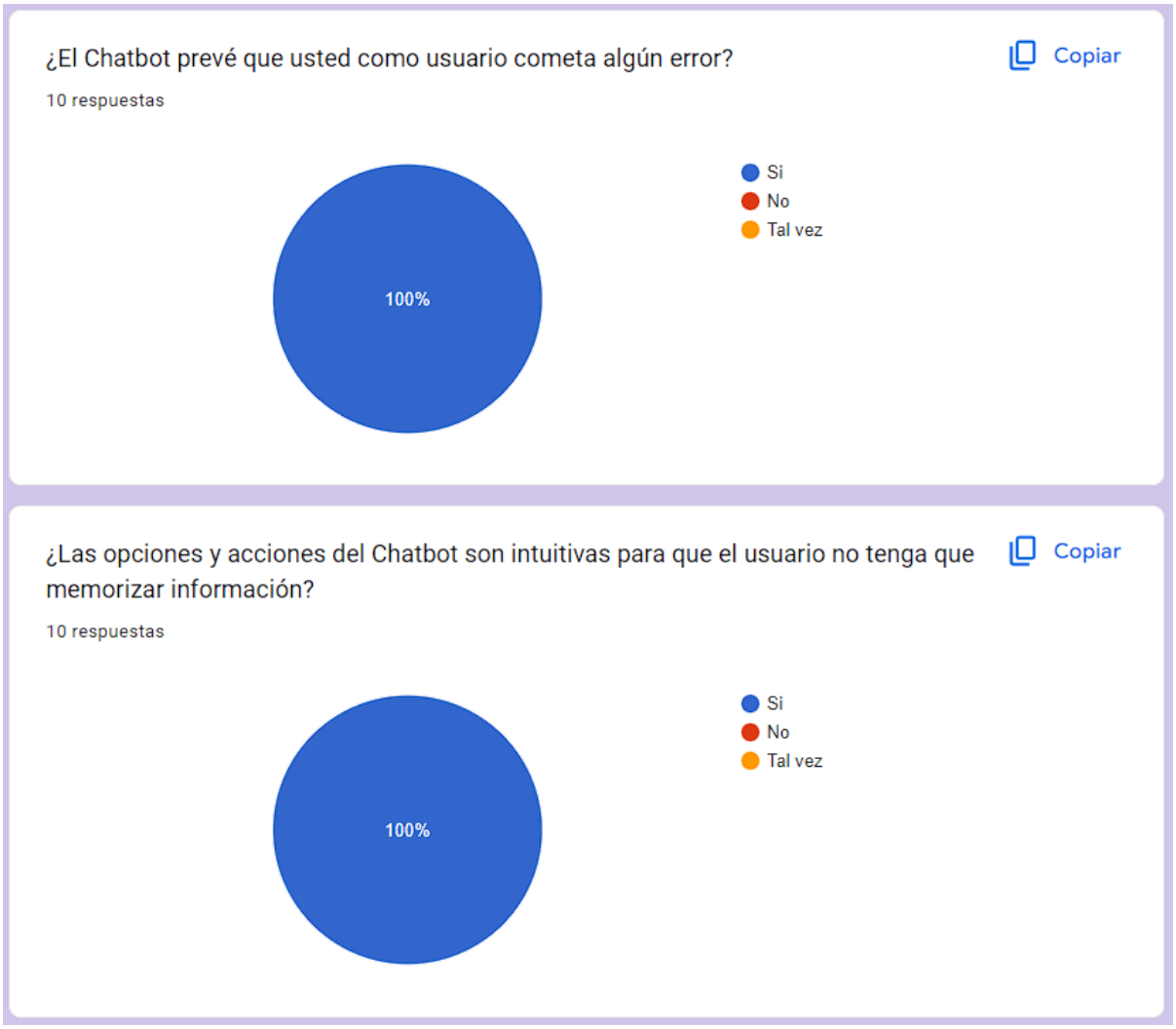


Figura 7.7: Encuesta Usuario Tendero Parte IV.

En esta sección, el encuestado responde a las preguntas número 7 y 8, las cuales hacen referencia a que el Chatbot debe ser intuitivo y fácil de usar, tanto para usuarios básicos, como avanzados. Asimismo, se debe velar para que el Chatbot no contenga información innecesaria para el usuario. Evidenciando en la Figura siguiente, que el 100 % de los encuestados opinaron que el Chatbot cumplía totalmente con ambos principios de usabilidad.

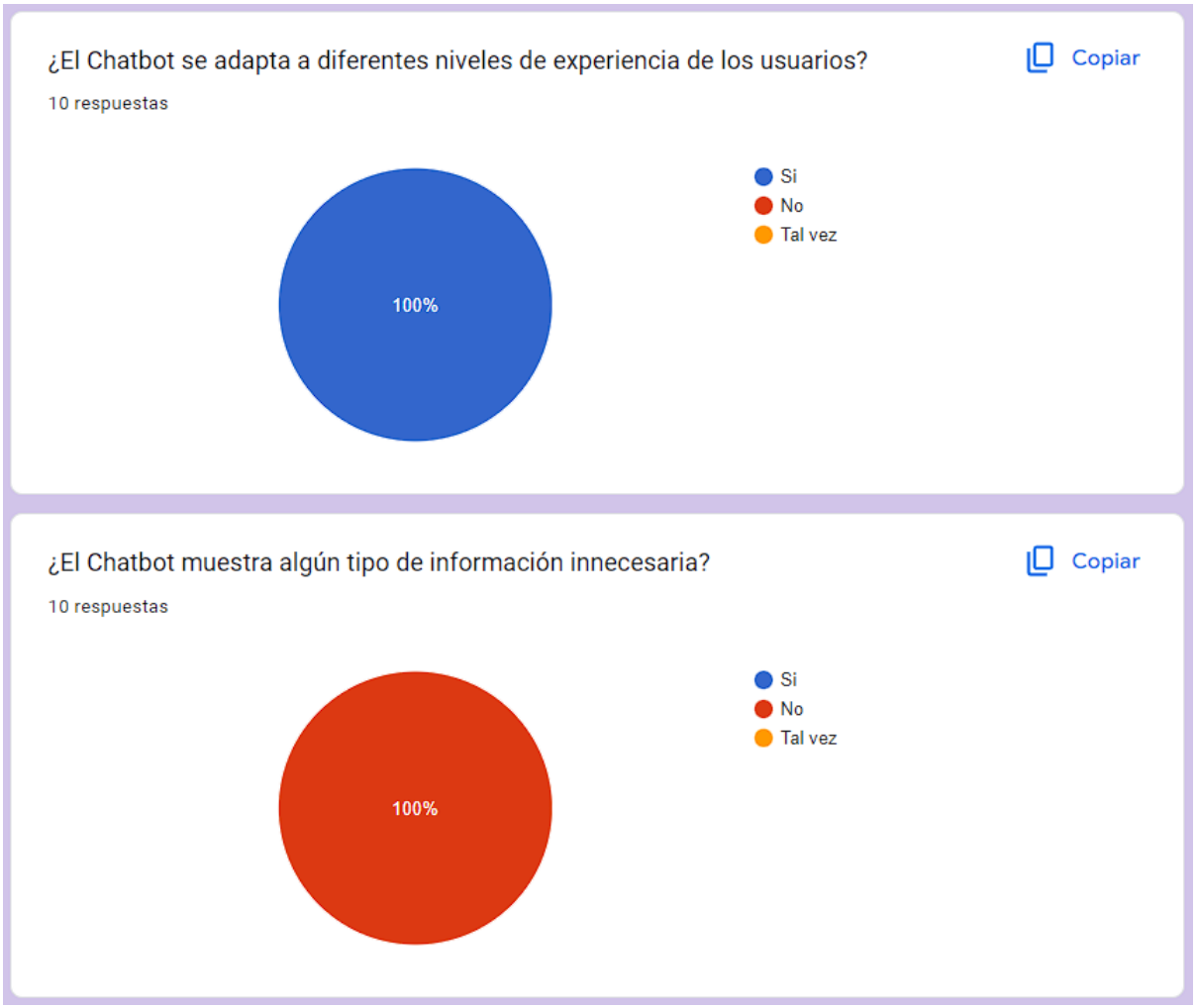


Figura 7.8: Encuesta Usuario Tendero Parte V.

En esta sección, el encuestado responde a las preguntas número 9 y 10, las cuales hacen referencia a que los mensajes de error o alertas del Chatbot, se deben mostrar en un lenguaje claro, indicando de forma precisa el problema, y sugiriendo una solución al respecto. Asimismo, el Chatbot debe contar con ayudas en cuanto a su funcionamiento, como lo son las instrucciones y documentación adicional. Evidenciando en la Figura siguiente, que el 100 % de los encuestados opinaron que el Chatbot cumplía totalmente con ambos principios de usabilidad.

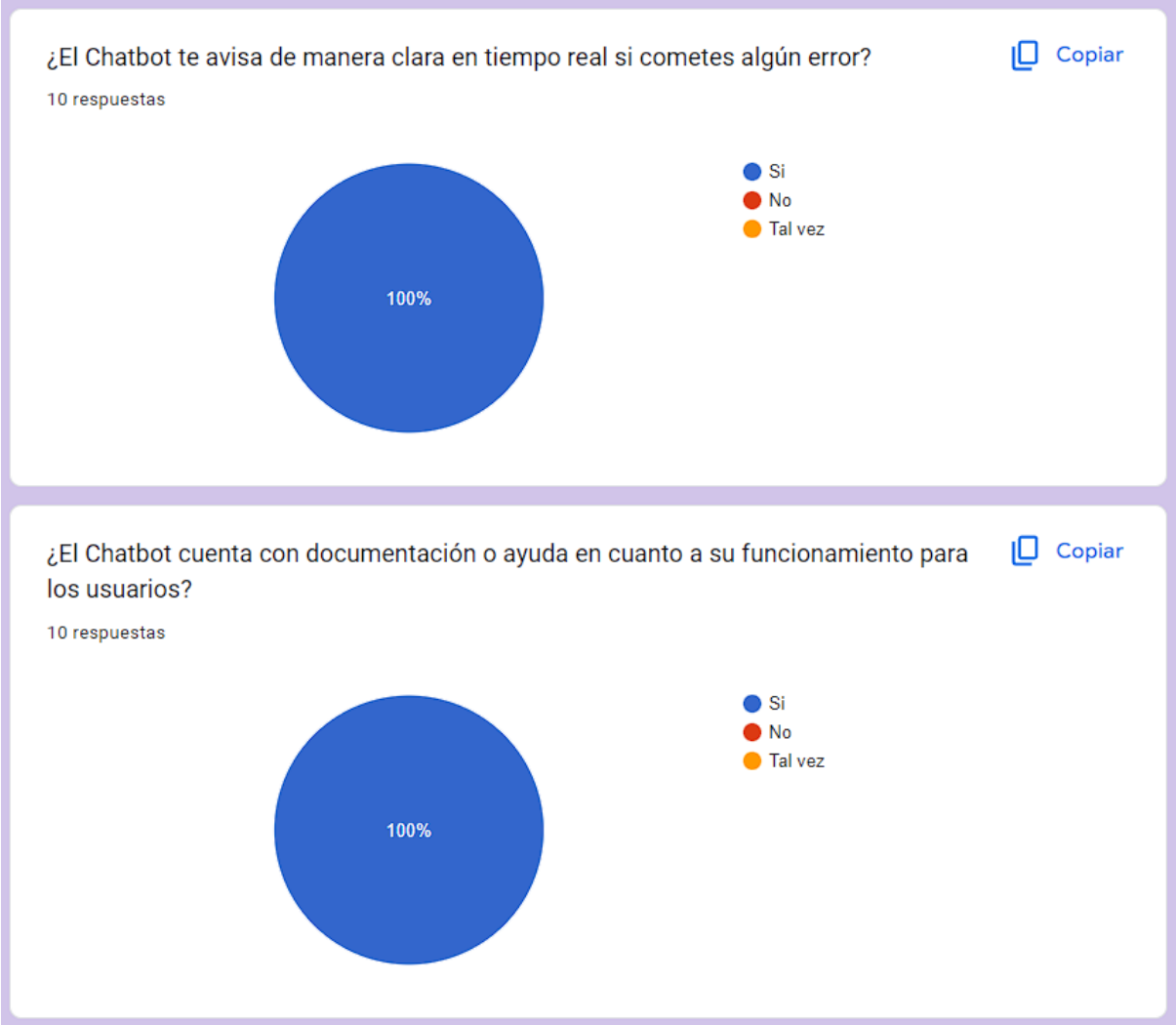


Figura 7.9: Encuesta Usuario Tendero Parte VI.

En esta sección, el encuestado podía escribir sugerencias o comentarios que pueden ser valiosos para mejorar el funcionamiento del Chatbot a futuro.

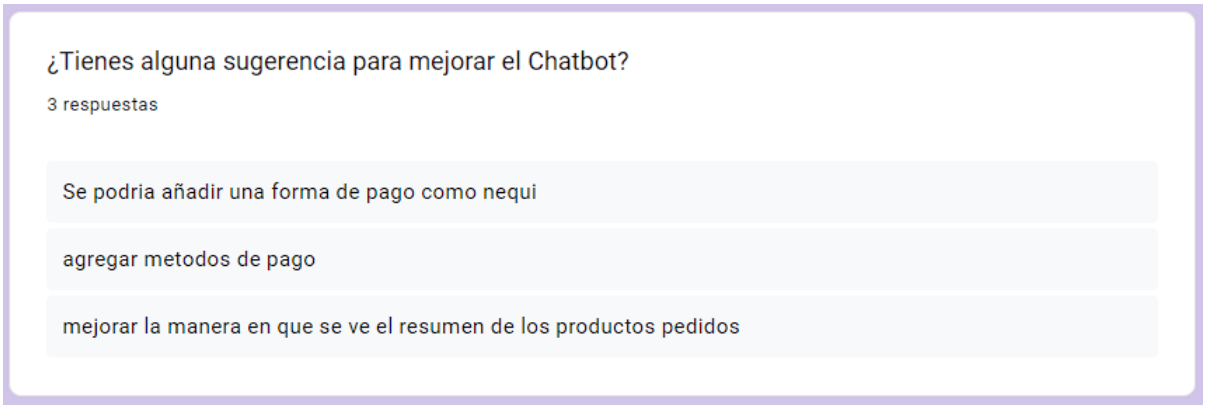


Figura 7.10: Encuesta Usuario Tendero Parte VII.

7.2.2. Resultados Usuario Proveedor

En esta sección, el encuestado responde a las preguntas número 1 y 2, las cuales hacen referencia a los principios número 1 y número 2 de usabilidad, que indican que el usuario siempre debe mantenerse informado de su estado o de lo que está sucediendo, y, que se debe utilizar un lenguaje con palabras y expresiones familiares para el usuario. Evidenciando en la Figura siguiente, que el 100 % de los encuestados opinaron que el Chatbot cumplía totalmente con ambos principios de usabilidad.

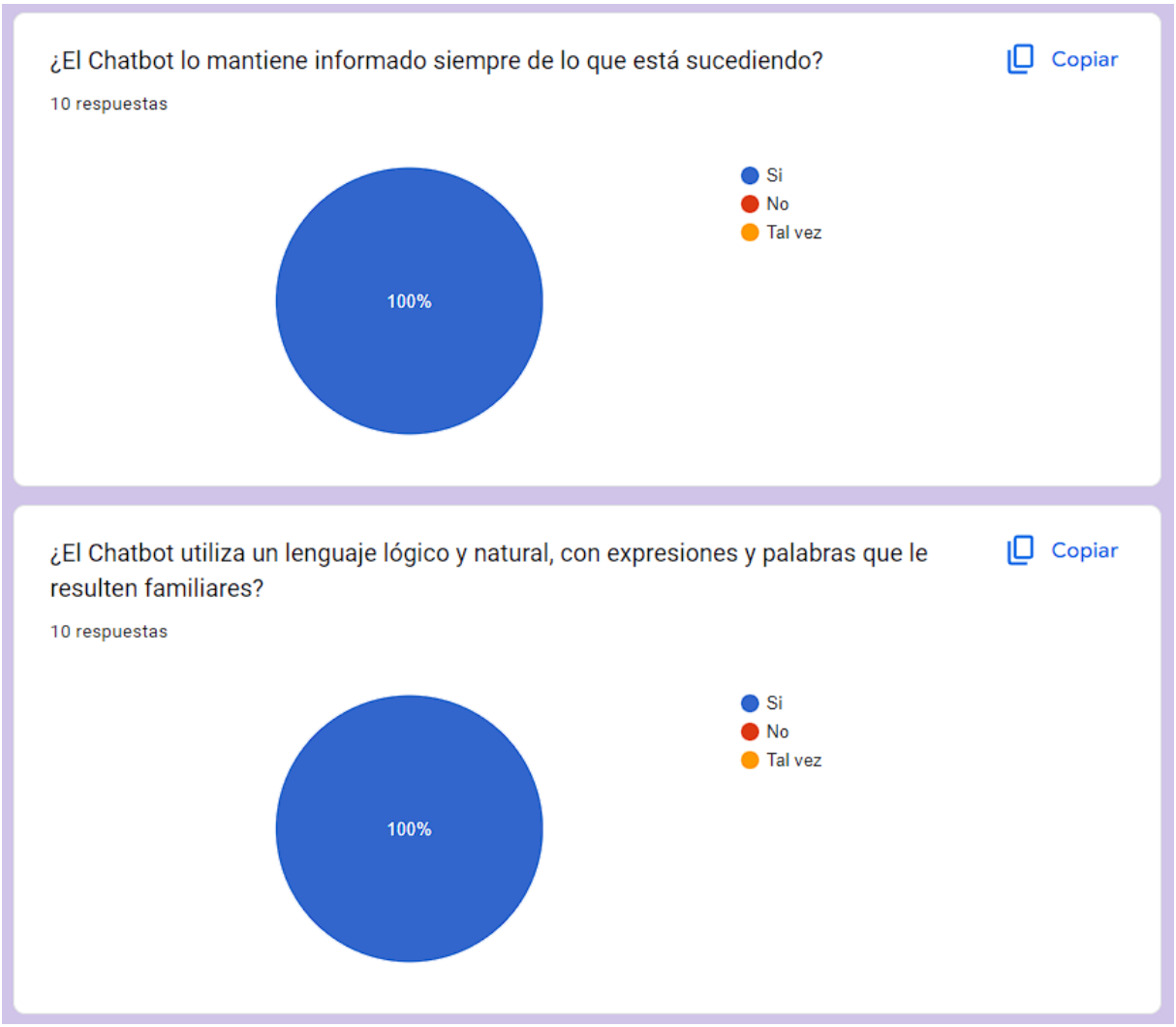


Figura 7.11: Encuesta Usuario Proveedor Parte I.

En esta sección, el encuestado responde a las preguntas números 3 y 4, las cuales hacen referencia a que un usuario debe poder deshacer o repetir una acción previamente realizada, y, que el Chatbot debe mantener ciertas convenciones lógicas en cuanto a su presentación y diseño. Evidenciando en la Figura siguiente, que el 100 % de los encuestados opinaron que el Chatbot cumplía totalmente con ambos principios de usabilidad.

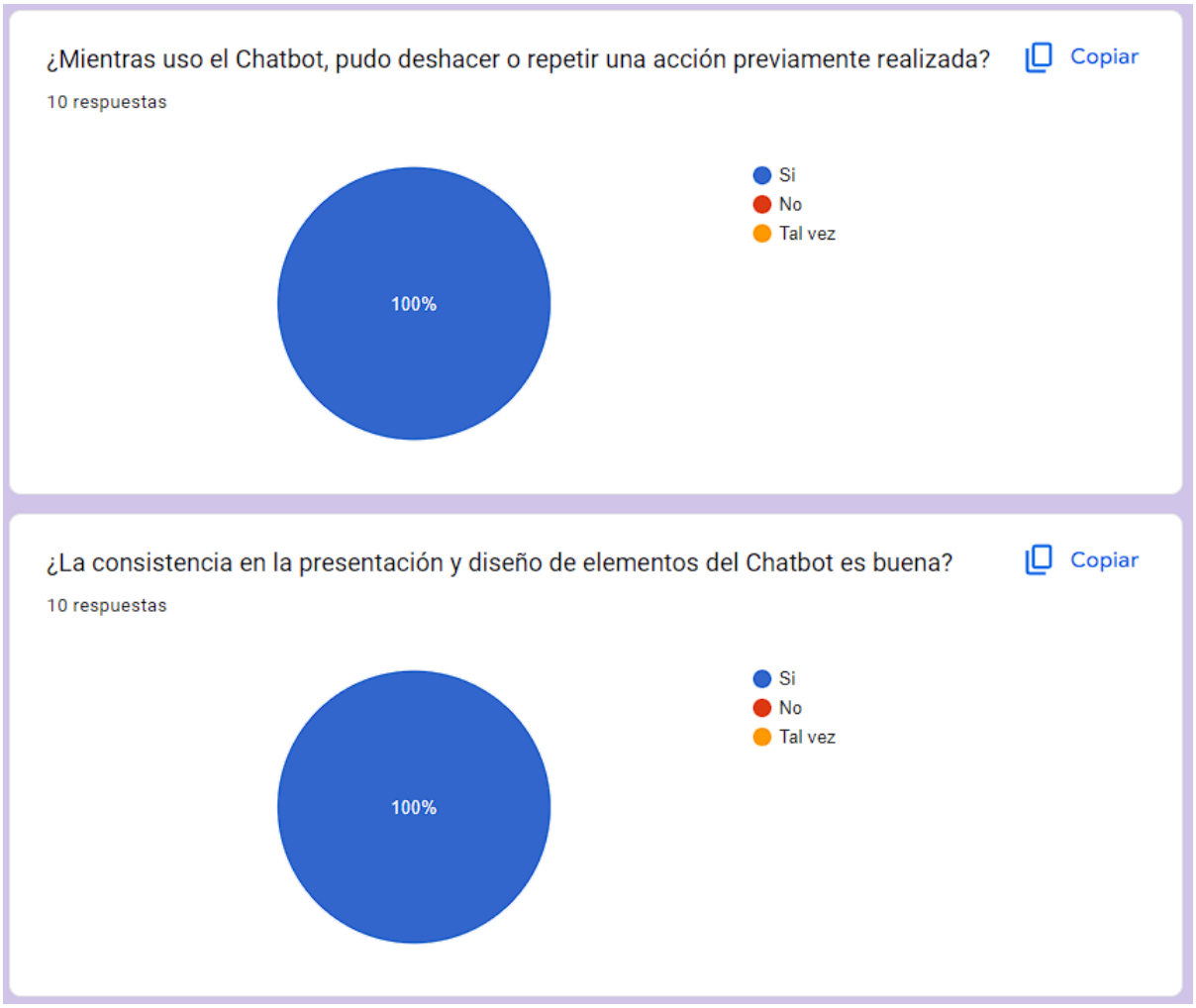


Figura 7.12: Encuesta Usuario Proveedor Parte II.

En esta sección, el encuestado responde a las preguntas números 5 y 6, las cuales hacen referencia a que el Chatbot debe estar preparado para ayudar al usuario en caso tal de que éste cometa un error. Asimismo, que las opciones y acciones del Chatbot deben ser intuitivas para que el usuario no tenga que memorizar información mientras hace uso de este. Evidenciando en la Figura siguiente, que el 100 % de los encuestados opinaron que el Chatbot cumplía totalmente con ambos principios de usabilidad.

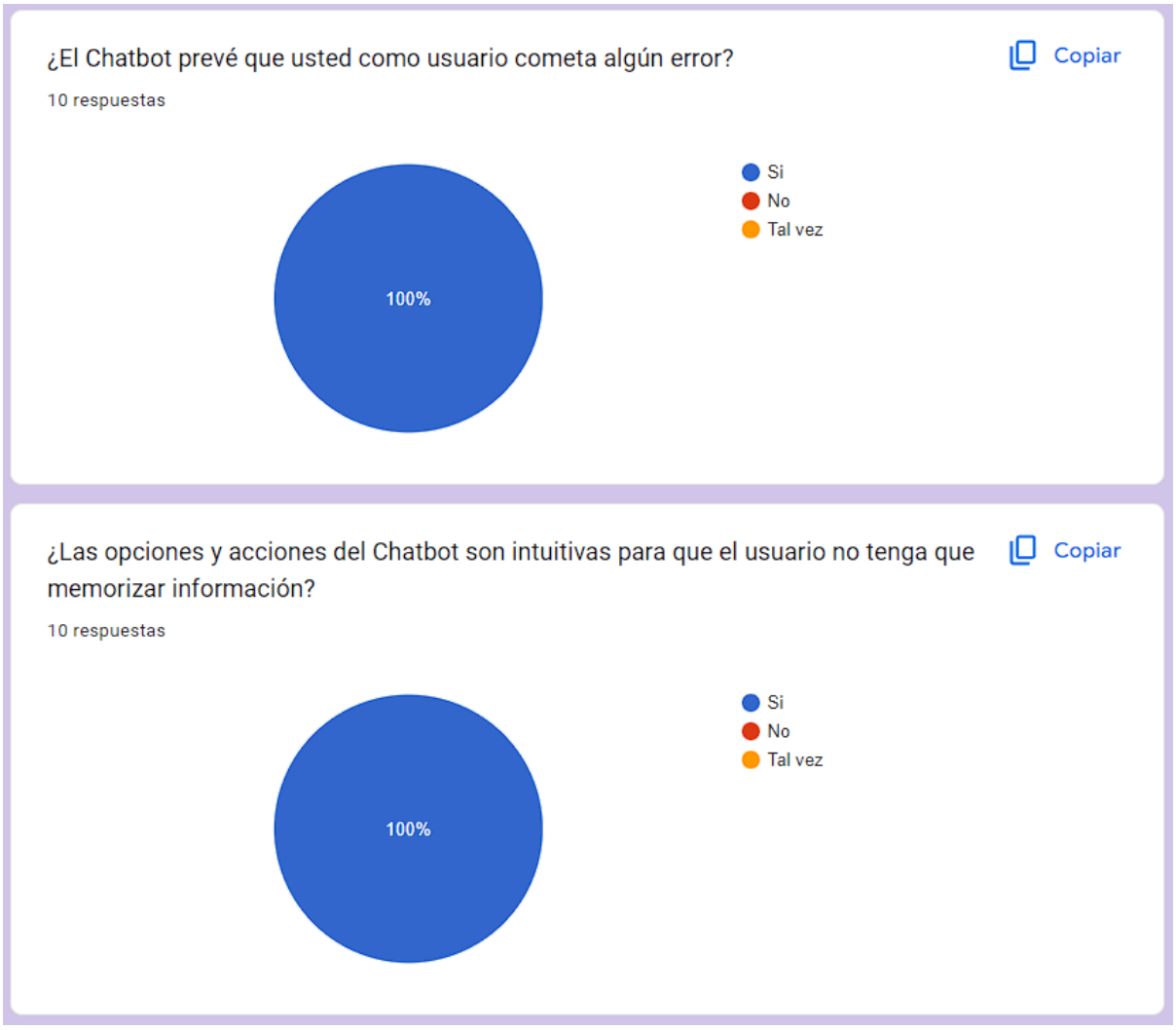


Figura 7.13: Encuesta Usuario Proveedor Parte III.

En esta sección, el encuestado responde a las preguntas número 7 y 8, las cuales hacen referencia a que el Chatbot debe ser intuitivo y fácil de usar, tanto para usuarios básicos, como avanzados. Asimismo, se debe velar para que el Chatbot no contenga información innecesaria para el usuario. Evidenciando en la Figura siguiente, que el 100 % de los encuestados opinaron que el Chatbot cumplía totalmente con ambos principios de usabilidad.

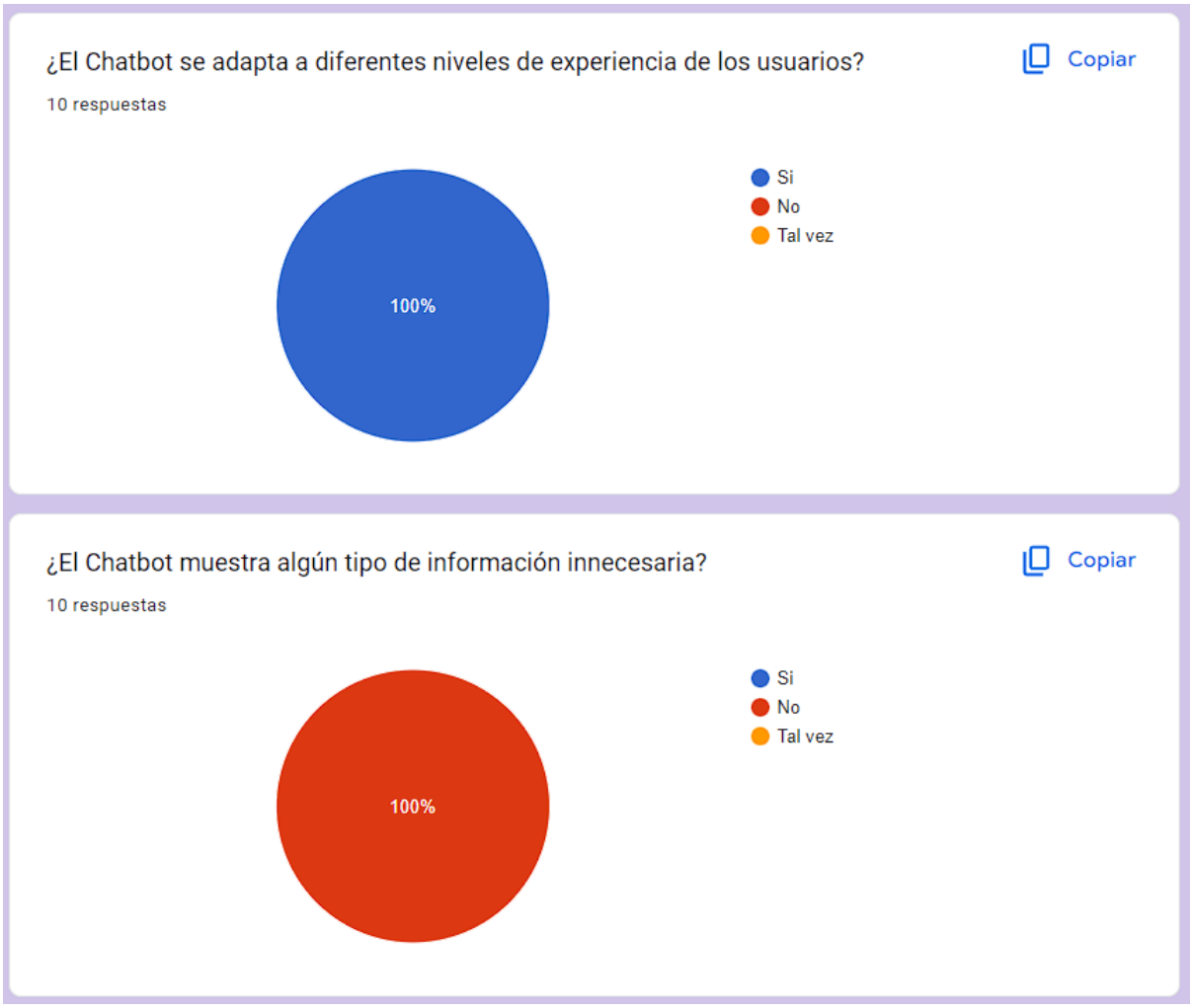


Figura 7.14: Encuesta Usuario Proveedor Parte IV.

En esta sección, el encuestado responde a las preguntas número 9 y 10, las cuales hacen referencia a que los mensajes de error o alertas del Chatbot, se deben mostrar en un lenguaje claro, indicando de forma precisa el problema, y sugiriendo una solución al respecto. Asimismo, el Chatbot debe contar con ayudas en cuanto a su funcionamiento, como lo son las instrucciones y documentación adicional. Evidenciando en la Figura siguiente, que el 100 % de los encuestados opinaron que el Chatbot cumplía totalmente con ambos principios de usabilidad.

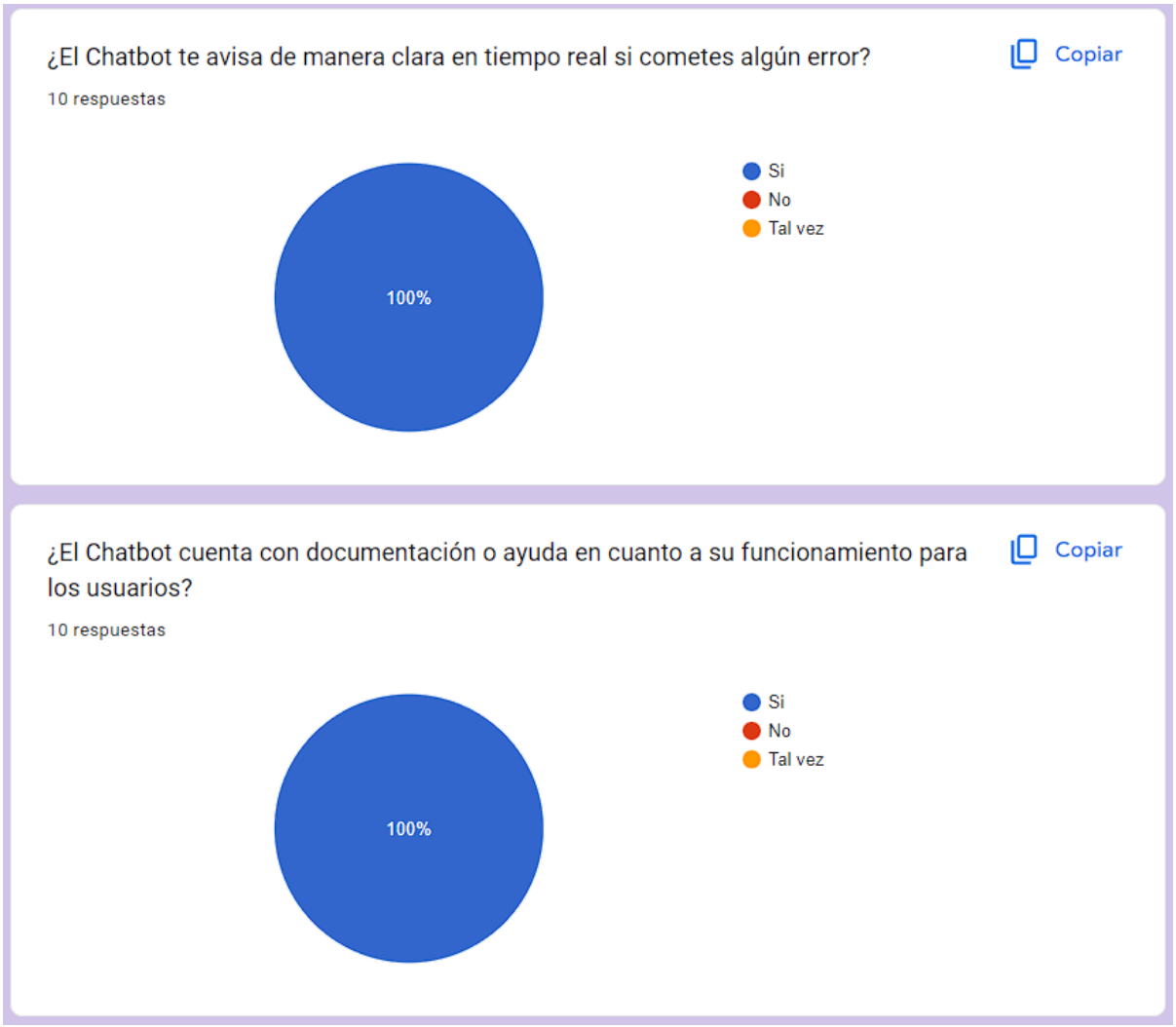


Figura 7.15: Encuesta Usuario Proveedor Parte V.

7.2.3. Comentarios y Conclusiones

Se encuestó a un total de 20 personas (10 tenderos y 10 proveedores), para esto se abordó a cada uno de ellos y se les explicó, de manera sencilla, el propósito y funcionamiento del Chatbot, enmarcado como un proyecto de grado. Del mismo modo, se pidieron comentarios y críticas constructivas a modo de retroalimentación; la encuesta consta de una serie de preguntas para evaluar los principios de usabilidad de Jakob Nielsen y también un campo para comentarios y críticas de los usuarios; los resultados de la encuesta mostraron un alto nivel de apoyo y satisfacción por parte de los encuestados.

Dentro de los comentarios recibidos en las encuestas, de manera formal, y como retroalimentación, los encuestados Tenderos, después de ver el funcionamiento del Chatbot, manifestaron, como se puede apreciar en la Figura 7.7, que sería bueno al momento de realizar el pedido, en un futuro poder implementar un método de pago para poder cancelar el pedido directamente al Proveedor y que este simplemente lo despache y lo haga llegar a la dirección del tendero. Asimismo, se dio la recomendación de mejorar la manera en que se ve el resumen de los productos pedidos por parte del Tendero en el Chatbot de Telegram.

Por parte de los encuestados Proveedores, se obtuvo la sugerencia de implementar en el carrito de la Web App o al momento de seleccionar un producto que se le muestra al Tendero, una manera de poder escribir la cantidad requerida de dicho producto, para que no se tenga que presionar muchas veces un botón para añadir cierta cantidad de productos requeridos por el Tendero.

Capítulo 8

Evidencias

8.1. Proceso Levantamiento de Requisitos

Para poder realizar el levantamiento de los requisitos funcionales y no funcionales que requerían los stakeholders (Proveedores y Tenderos) para crear el Chatbot, se utilizaron distintas herramientas como lo son las lluvias de ideas, entrevistas, observación, diagrama de casos de uso y prototipado del Chatbot. Las cuales sirvieron para comprender mejor su cadena de abastecimiento, su entorno y las variables relevantes que se deben tener en cuenta para la toma de decisiones.

Inicialmente, se identificó la población objetivo y se buscó la manera de empatizar con alguna persona conocida que se encontrará en este gremio de Proveedores y Tenderos, es así cómo se llegó a conocer a Cristian Alejandro Gómez, propietario y encargado de la empresa Distri Romel S.A.S. ubicada en el barrio popular, la cual se encarga de distribuir distintos productos no perecederos a las tiendas de barrio en distintos sectores en la ciudad de Cali como se puede apreciar en el mapa de la siguiente imagen.

Figura 8.1: Evidencias foto mapa Distri Romel.

Asimismo, Cristian narró el proceso que ellos realizan para abastecer las distintas tiendas de

barrio. Para este proceso ellos inicialmente le pagan a un intermediario el cual envía personas para que vayan ofreciendo sus productos a distintos sectores de la ciudad tienda por tienda con un catálogo de productos suyo impreso (Véase la figura 8.2) para posteriormente tomar los pedidos a mano y hacerlos llegar al proveedor, en este caso Distri Romel, los cuales se encargan de transcribir dichos pedidos a un Excel para llevar un registro y de preparar todo para despachar dichos pedidos al día siguiente a cada una de las tiendas, lo cual de acuerdo a los comentarios de Cristian resulta ser un proceso tedioso y después de tener una conversación con él se llegó a la conclusión de que estos procesos se podrían mejorar con la implementación de alguna solución tecnológica como puede ser una aplicación Web, Móvil o un Chatbot.

Figura 8.2: Evidencias logo y catalogo Distri Romel.

Es aquí donde empieza el proceso de levantamiento de requisitos para el cual inicialmente se acuerda una cita con Cristian, en su bodega en el barrio popular, en la cual por medio de una lluvia de ideas, análisis y observación, se logran obtener distintos requisitos que se deben tener en cuenta en el momento de la realización de dicha solución tecnológica.

Posteriormente, aprovechando que en el mismo barrio hay distintas tiendas de barrio a las cuales él les surte y otras bodegas, se procede a realizar distintas entrevistas a los tenderos y proveedores, en las cuales se pudo apreciar necesidades muy importantes que se deben tener en cuenta para diferenciar que tipo de solución tecnológica se debía implementar, ya que se pudo apreciar que la mayoría de tenderos eran adultos mayores, que contaban con un plan prepago en su celular y que por lo general solo sabían realizar llamadas con este o usar plataformas de mensajería instantánea como WhatsApp.

Después de esto se realizó un proceso de investigación para buscar distintas aplicaciones exis-

tentes como lo son SurtiApp [6] y Chiper [7], las cuales son tecnologías que han intentado aportar soluciones para establecer una comunicación directa entre proveedores y tiendas de barrio, pero que no brindan un servicio personalizado y adaptado a las necesidades de las tiendas de barrio típicas colombianas, ya que no se tiene en cuenta que estas personas por lo general no cuentan con los recursos necesarios para acceder a un plan de datos postpago en su celular, mediante el cual puedan acceder a una aplicación web móvil que se desarrolle.

Es por esto, que se descartaron las opciones de crear una aplicación web o móvil como solución para este problema de comunicación que se tiene entre los Proveedores y los Tenderos, quedando como única opción la creación de un Chatbot, el cual de acuerdo a la investigación realizada previamente, puede funcionar en distintas plataformas de mensajería instantánea como lo son WhatsApp y Telegram, a las cuales se puede tener acceso por distintos operadores al momento de realizar una recarga prepago.

Después de realizar un análisis y comparación entre las plataformas de mensajería instantánea, se llega a la conclusión de utilizar la API de Telegram para desarrollar el Chatbot, esto debido a que es de código abierto, es popular, muy similar a WhatsApp y cuenta con una funcionalidad que puede ser clave la cual es la creación de una Web App que se puede ejecutar dentro de la misma conversación con un Chatbot, que puede ir guiando por medio de una interfaz a los tenderos para poder realizar su pedido exitosamente.

Teniendo en cuenta lo anterior, se crea un diagrama de casos de uso (Véase la figura 3.1) mediante el cual se pretende modelar las distintas acciones que va a poder realizar tanto el Usuario Proveedor como el Usuario Tendero, teniendo en cuenta dicho diagrama se sacan más requisitos funcionales y no funcionales que se deben tener en cuenta para almacenar datos, soportar las comunicaciones con el Chatbot y demás.

Asimismo, se crea un prototipo inicial haciendo uso de la herramienta “balsamiq mockups”, de las distintas vistas con las que se pretende que va a contar el Chatbot y en donde se muestra las distintas funcionalidades con las que se piensa que este va a contar de acuerdo con los requisitos obtenidos hasta el momento. Para ver más información sobre los prototipos o ”mockups”que se han desarrollado durante esta fase, se recomienda revisar la sección 10.2 del Capítulo de Anexos, la cual contiene documentación de todos los prototipos que se han desarrollado antes de empezar a trabajar en el software.

Una vez se habían desarrollado dichos prototipos del Chatbot en Telegram, se decidió mostrar los mismos a los stakeholders con el fin de poder recibir una primera retroalimentación, como se puede ver en la sección 4.5.1, en la cual se obtuvieron algunas funcionalidades más por implementar que resultaron ser cruciales para poder continuar desarrollando el Chatbot de acuerdo con sus necesidades.

Finalmente, luego de todo el proceso de desarrollo del software del Chatbot, se decide realizar pruebas de usabilidad de acuerdo a los 10 principios de usabilidad de Jakob Nielsen, por medio de una encuesta a los distintos stakeholders, como se puede evidenciar en la sección de pruebas de usabilidad en el capítulo 7, para validar que se cumpla dicho principio de usabilidad en el Chatbot desarrollado de acuerdo a sus necesidades. Por medio de esta encuesta también se obtienen algunas sugerencias que se podrán implementar como trabajos futuros. A continuación se muestran algunas

imágenes que sirven como evidencia de las distintas visitas y salidas que se realizaron en todo este proceso mencionado en el capítulo:

Figura 8.3: Evidencias pruebas con Tenderos y Proveedores.

8.2. Proceso Metodología SCRUM

Para el desarrollo del proyecto se hizo uso de la metodología ágil Scrum, mencionada anteriormente en la sección 2.1.2 del marco teórico, debido a su popularidad a nivel profesional y la familiaridad que se tiene con este marco de trabajo, dado que a lo largo de la carrera se ha utilizado Scrum para realizar múltiples proyectos y actividades que han permitido acogernos al enfoque del desarrollo ágil. Por esto, se conoce a buen nivel de detalle los roles, eventos y herramientas necesarias para lograr aplicar Scrum a un proyecto de desarrollo de software, logrando trabajar de manera colaborativa, buscando siempre obtener el mejor resultado posible para los proyectos propuestos.

Para llevar a cabo el proyecto bajo esta metodología, se efectuaron finalmente 7 sprints o iteraciones que duraban un periodo de tiempo de 30 días, en los cuales se iban definiendo las distintas actividades que se iban a ir realizando para dicho sprint, lo cual permitió ir llevando un seguimiento y control de las mismas, para poder cumplir con los requisitos establecidos previamente y lograr completar el proyecto de manera satisfactoria.

A continuación, se mostrarán los distintos sprints que se llevaron a Cabo:

ACTIVIDADES	FECHA INICIO	FECHA ENTREGA	DURACION (DIAS)	ESTADO	DETALLE / COMENTARIO
SPRINT 1	15-nov-22	15-dic-22	30		
Primer acercamiento a uno de los stakeholders	16-nov-22	16-nov-22	1	Terminada	
Buscar e identificar las diferentes plataformas o aplicaciones que poseen características similares a la propuesta planteada.	17-nov-22	25-nov-22	8	Terminada	Se encuentra plataformas como Surtiapp y Chipper
Realizar consultas e investigaciones pertinentes para verificar procesos de las funciones que realizan dichas plataformas o aplicaciones.	26-nov-22	4-dic-22	8	Terminada	
Salida de campo para dialogar con stakeholders e identificar sus necesidades	5-dic-22	5-dic-22	1	Terminada	Se identifica falta de recursos para el plan de datos
Prototipado	6-dic-22	7-dic-22	2	Terminada	
Enseñar prototipado a los stakeholders para retroalimentacion	8-dic-22	8-dic-22	1	Terminada	
Identificar los casos de uso y escenarios de conversación.	9-dic-22	9-dic-22	1	Terminada	
Establecer objetivos, alcance y requisitos del Chatbot.	9-dic-22	12-dic-22	3	Terminada	
Documentar el avance realizado en el proyecto de manera escrita.	9-dic-22	14-dic-22	4	Terminada	
Revision de avances y reunion con el Director del Trabajo de Grado	15-dic-22	15-dic-22	1	Terminada	

Figura 8.4: Evidencia Metodología Scrum Sprint I.

Como se puede apreciar en la Figura anterior, el desarrollo del primer sprint resulta ser de suma importancia, ya que es aquí donde se logra identificar las distintas necesidades con las que cuenta la población objetivo de estudio y se empieza a investigar al respecto, para saber si existen algunas soluciones tecnológicas que les puedan ayudar. Asimismo, en este sprint se logra identificar la falta de recursos que se tiene en dicha población con respecto a su plan de datos y se descubre el porqué no se puede implementar una solución tecnológica convencional como lo es una aplicación web o móvil.

Es por esto, que se plantea hacer un prototipado de un Chatbot el cual se muestra a la población objetivo y se obtienen recomendaciones relevantes como el de incluir un botón de contacto para que los tenderos se pudieran contactar con sus proveedores, un video en la sección de ayuda y por parte de los proveedores que se pudiera conectar el chatbot con su base de datos de Excel para que estos pudieran manipular mejor sus datos. Gracias a esto, se logra realizar dichos ajustes al prototipo y se logra identificar los casos de uso, se establecen objetivos, alcances y la mayoría de requisitos que se deben tener en cuenta para la realización del Chatbot.

ACTIVIDADES	FECHA INICIO	FECHA ENTREGA	DURACION (DIAS)	ESTADO	DETALLE / COMENTARIO
SPRINT 2	15-dic-22	15-ene-23	30		
Crear un diseño de conversación y flujo de interacción del Chatbot.	15-dic-22	17-dic-22	2	Terminada	
Definir las respuestas del Chatbot para diferentes escenarios.	18-dic-22	20-dic-22	2	Terminada	
Diseñar el esquema de menús y opciones para una navegación clara.	21-dic-22	23-dic-22	2	Terminada	
Diseñar la arquitectura de software que mejor se acomode a los requisitos planteados.	24-dic-22	31-dic-22	7	Terminada	
Analizar los modelos y patrones usados para garantizar una mejor comunicación entre los stakeholders.	1-ene-23	9-ene-23	7	Terminada	
Diseñar un modelo de datos que recoja todas las entidades y atributos planteados en los requisitos.	5-ene-23	10-ene-23	5	Terminada	
Documentar el avance realizado en el proyecto de manera escrita.	10-ene-23	14-ene-23	4	Terminada	
Revision de avances y reunion con el Director del Trabajo de Grado	15-ene-23	15-ene-23	1	Terminada	

Figura 8.5: Evidencia Metodología Scrum Sprint II.

Para la realización del segundo sprint, se plantean actividades que están más ligadas al proceso del diseño del Chatbot que se piensa desarrollar, teniendo en cuenta los requisitos obtenidos previamente, se realiza un diagrama de flujo para representar cómo va a ser el flujo de la conversación del Chatbot, el cual permite comprender mejor cómo se deben definir las respuestas del mismo, sus

opciones y se logra definir una arquitectura MVC que se adapte con los requisitos de este.

ACTIVIDADES	FECHA INICIO	FECHA ENTREGA	DURACION (DIAS)	ESTADO	DETALLE / COMENTARIO
SPRINT 3	15-ene-23	15-feb-23	30		
implementar los diseños previos teniendo en cuenta los requisitos establecidos.	15-ene-23	20-ene-23	5	Terminada	
Desarrollar el modelo de datos.	21-ene-23	26-ene-23	5	Terminada	
Analizar y seleccionar las herramientas y tecnologías para el desarrollo del Chatbot.	27-ene-23	3-feb-23	7	Terminada	
Configurar el entorno de desarrollo para el Chatbot en Telegram.	7-feb-23	10-feb-23	3	Terminada	Selección de la librería PyTelegramBotApi
Establecer la conexión con la plataforma de Telegram.	11-feb-23	13-feb-23	2	Terminada	
Implementar las capacidades básicas de respuesta del Chatbot.	11-feb-23	14-feb-23	3	Terminada	
Documentar el avance realizado en el proyecto de manera escrita.	10-feb-23	14-feb-23	4	Terminada	
Revision de avances y reunion con el Director del Trabajo de Grado	15-feb-23	15-feb-23	1	Terminada	

Figura 8.6: Evidencia Metodología Scrum Sprint III.

Para este tercer sprint, se crea inicialmente un modelo relacional con el objetivo de comprender mejor cómo se van a organizar los datos en las distintas tablas que va a comprender la base de datos del Chatbot. Asimismo, se analizan y seleccionan, las distintas herramientas tecnológicas que se van a utilizar para el desarrollo de este y se comienza a configurar el entorno básico del Chatbot.

En esta última actividad, se encontró un gran inconveniente, ya que inicialmente se tenía pensado realizar el Chatbot haciendo uso de la herramienta Dialogflow de Google, pero al momento de hacer la implementación de la misma junto con Python se pudo ver que esta tenía ciertas limitaciones que impedían seguir con el proceso de desarrollo planteado, es por esto, que se decidió hacer uso de la librería PyTelegramBotApi, la cual permite establecer la comunicación con la API de Telegram sin limitaciones y es mucho más personalizable para poder implementar las capacidades básicas de respuesta del Chatbot.

ACTIVIDADES	FECHA INICIO	FECHA ENTREGA	DURACION (DIAS)	ESTADO	DETALLE / COMENTARIO
SPRINT 4	15-feb-23	15-mar-23	30		
Desarrollar el código correspondiente del Backend del Chatbot.	15-feb-23	23-feb-23	8	Terminada	
Desarrollar y probar las funcionalidades basicas del Chatbot.	24-feb-23	1-mar-23	5	Terminada	
Implementar la lógica para el manejo de solicitudes y respuestas.	1-mar-23	6-mar-23	5	Terminada	
Desarrollar el código correspondiente del Frontend del Chatbot.	7-mar-23	12-mar-23	5	Terminada	Esta parte corresponde a la creacion en React de la Web App
Pruebas independientes para verificar el funcionamiento del Frontend y el Backend	10-mar-23	12-mar-23	2	Terminada	
Documentar el avance realizado en el proyecto de manera escrita.	10-mar-23	14-mar-23	4	Terminada	
Revision de avances y reunion con el Director del Trabajo de Grado	15-mar-23	15-mar-23	1	Terminada	

Figura 8.7: Evidencia Metodología Scrum Sprint IV.

Como se puede apreciar en la Figura anterior, para el cuarto sprint se desarrollaron actividades que se encontraban relacionadas con la parte del desarrollo como tal, en la cual se crearon las funcionalidades básicas del Chatbot y se implementa la lógica necesaria en el back, para el manejo de las solicitudes y respuestas de este. Asimismo, se desarrolló la parte del front que va a ser de suma importancia para poder guiar al usuario en la realización de su pedido por medio de la interfaz y por último se realizan pruebas independientes con algunos casos de prueba, para probar la funcionalidad tanto del Front End como del Back end.

ACTIVIDADES	FECHA INICIO	FECHA ENTREGA	DURACION (DIAS)	ESTADO	DETALLE / COMENTARIO
SPRINT 5	15-mar-23	15-abr-23	30		
Investigacion de las plataformas para realizar el despliegue del Front, Back y Base de datos	15-mar-23	23-mar-23	8	Terminada	Ngrok, Netlify, AWS
Integrar las API y servicios necesarios para el funcionamiento del Chatbot.	24-mar-23	3-abr-23	10	Terminada	Ngrok no sirve, se cambia por Localhost
Realizar correcciones y pruebas de integracion pertinentes.	4-abr-23	11-abr-23	7	Terminada	
Documentar el avance realizado en el proyecto de manera escrita.	11-abr-23	14-abr-23	4	Terminada	
Revision de avances y reunion con el Director del Trabajo de Grado	15-abr-23	15-abr-23	1	Terminada	

Figura 8.8: Evidencia Metodología Scrum Sprint V.

Para el quinto sprint, se tenía propuesto lograr la integración de las API y de los servicios

necesarios para el correcto funcionamiento del Chatbot. Es por esto, que se realiza la investigación pertinente para evaluar las distintas maneras en las que se puede desplegar tanto el front como el back y la base de datos relacional.

Inicialmente, se llegó a la conclusión de utilizar AWS para desplegar la base de datos relacional en su servicio RDS, Netlify para desplegar la página estática correspondiente al Front y por último Ngrok para desplegar el Back, pero para esta última parte, se tuvo inconvenientes, puesto que Ngrok no permitía generar una URL que fuera estática y que tuviera el protocolo de transferencia de hipertexto seguro (HTTPS). Por lo cual se decidió cambiar Ngrok por LocalTunnel, la cual sí permite generar una URL estática a través de subdominios y esta contaba con el protocolo HTTPS que resulta crucial para poder conectarse a la API de Telegram, además esta permite exponer el servidor local creado con Flask de una manera muy simple en dicho subdominio creado.

Finalmente, se logra realizar la integración de los distintos componentes y se realizan las correcciones necesarias, como por ejemplo la declaración asíncrona de los endpoints creados en Flask para que cada una de las peticiones se pueda manejar en un hilo independiente

ACTIVIDADES	FECHA INICIO	FECHA ENTREGA	DURACION (DIAS)	ESTADO	DETALLE / COMENTARIO
SPRINT 6	15-abr-23	15-may-23	30		
Implementar mecanismos para manejar errores en la conversacion con el Chatbot.			5	Terminada	
Realizar y corregir pruebas funcionales al Chatbot.			10	Terminada	Se crean distintos casos de prueba
Realizar pruebas de usabilidad a los distintos stakeholders.			2	Terminada	Se tiene en cuenta los 10 principios de usabilidad de Jakob Nielsen
Analisis de las pruebas, conclusiones y trabajos futuros.			2	Terminada	
Documentar el avance realizado en el proyecto de manera escrita.			10	Terminada	
Revision de avances y reunion con el Director del Trabajo de Grado			1	Terminada	

Figura 8.9: Evidencia Metodología Scrum Sprint VI.

Para este sexto sprint, se propone implementar mecanismos para manejar errores dentro de la conversación del Chatbot, como lo son la validación de la entrada de los textos por parte del usuario, que sí corresponda al valor esperado. Asimismo, se realizan distintas pruebas funcionales después de la integración, basadas en casos de prueba que permiten validar su funcionalidad. También, se llevan a cabo pruebas de usabilidad a los distintos stakeholders basándose en los 10 principios de usabilidad de Jakob Nielsen, de las cuales se obtienen resultados muy satisfactorios y se logran sacar distintas conclusiones y trabajos futuros de acuerdo a la retroalimentación obtenida.

ACTIVIDADES	FECHA INICIO	FECHA ENTREGA	DURACION (DIAS)	ESTADO	DETALLE / COMENTARIO
SPRINT 7	15-may-23	15-jun-23	30		
Preparar la documentación y manual de usuario del Chatbot.	16-may-23	20-may-23	4	Terminada	
Documentar el avance realizado en el proyecto de manera escrita.	20-may-23	4-jun-23	15	Terminada	
Realizar correcciones y ajustes finales al documento.	5-jun-23	15-jun-23	10	Terminada	
Revision de avances y reunion con el Director del Trabajo de Grado	15-jun-23	15-jun-23	1	Terminada	

Figura 8.10: Evidencia Metodología Scrum Sprint VII.

En este último sprint, al igual que en los demás, se propone documentar el avance realizado hasta el momento y se crea un manual de usuario que es muy importante para que el usuario pueda comprender cómo puede interactuar con el Chatbot y como es el funcionamiento del mismo. Para finalmente disponer el tiempo restante para llevar a cabo los ajustes o correcciones necesarias al documento del trabajo de grado, para que este sea evaluado y aprobado.

Capítulo 9

Conclusiones y Trabajos Futuros

9.1. Conclusiones

Las conclusiones generales del proyecto se dan con base en el cumplimiento de los objetivos propuestos y el alcance de estos:

- Partiendo del hecho de que el objetivo general de este proyecto era desarrollar el prototipo de un Chatbot que permitiera mejorar la comunicación entre proveedores y tiendas de barrio, se puede decir que se logró el objetivo realizando el proceso indicado de selección e implementación de las distintas tecnologías nombradas en el documento; teniendo en cuenta las distintas necesidades que se presentaron por parte de los stakeholders, con el fin de poder apoyarlos para poder lograr una transición digital que permita impulsar su negocio.
- Las tecnologías empleadas para desarrollar el proyecto, como lo son el lenguaje de programación Python y su framework Flask, designado para la parte del Chatbot y el Backend, el framework React para el Frontend y PostgreSQL para la base de datos relacional, son tecnologías ampliamente utilizadas en el mundo laboral y resultan ser de gran ayuda a la hora de desarrollar un Chatbot que hace uso de una aplicación web. Puesto que estas tecnologías no solo permiten ahorrar tiempo en el proceso de desarrollo debido a su fácil implementación, sino que también reciben soporte de una gran comunidad de desarrolladores que contribuyen a través de múltiples librerías que extienden sus funcionalidades.
- Se entendió a profundidad la importancia del uso de metodologías ágiles para el desarrollo iterativo e incremental de un proyecto que fue evolucionando mientras pasaba el tiempo, de acuerdo a las necesidades y soluciones que se presentaban en el continuo seguimiento del proyecto. Logrando comprender cómo las metodologías de desarrollo son cruciales para organizar previamente las diferentes etapas que se van a realizar en un proyecto y poder tener planes de contingencia en caso tal de que alguna de las etapas no se ejecute de acuerdo a lo planeado anteriormente.
- Asimismo, gracias a que las pruebas realizadas estaban orientadas al cumplimiento de los 10 principios de usabilidad propuestos por Jakob Nielsen, se logró verificar el cumplimiento de la mayoría de los requisitos funcionales con satisfacción. Por lo tanto, se puede concluir que el Chatbot cumple con los 10 principios de usabilidad, de acuerdo con los resultados obtenidos por las encuestas. Del mismo modo, cabe aclarar que debido al enfoque que se le dio al proyecto, durante su ejecución, las pruebas se centraron más en los requisitos funcionales y no en los requisitos no funcionales.

9.2. Trabajos Futuros

- Como trabajo futuro, se pretende realizar las mejoras planteadas al Chatbot por los distintos encuestados en la sección 7.2.3 de Comentarios y Conclusiones en el capítulo de Pruebas. Decisión tomada para poder dar seguimiento al proceso de mejora continua del desarrollo del software, enfocado en este caso hacia la población objetivo para la que está destinado el Chatbot.

- Asimismo, se propone realizar la conexión de la base de datos con una herramienta de visualización de datos como lo es Tableau o Qlik Sense, con el fin de proporcionar una visibilidad más clara y profunda de los datos, lo que puede ayudar a los usuarios a tomar decisiones más informadas y eficaces, de manera más rápida y sencilla gracias a los gráficos o estadísticas que este tipo de herramientas suelen brindar a través de las consultas necesarias.

- Finalmente, para salir a producción con el Chatbot, se debe contar con un mayor presupuesto, el cual permita desplegar el Chatbot en un servidor a través de un proveedor de servicios en la nube como lo son AWS o Google Cloud Platform, los cuales cobran cierto precio por unidad por el número de solicitudes de texto o voz que sean procesadas por el Chatbot.

Bibliografía

[1] "Colombia en el Contexto Internacional", MinTIC. [Online]. Available at: https://mintic.gov.co/portal/vivedigital/612/w3-article-1515.html.

[2] "Más de 11.000 tiendas de barrio podrían desaparecer en Colombia, según Fenalco", Semana, 2021. [Online]. Available at: https://www.semana.com/economia/empresas/articulo/mas-de-11000-tiendas-de-barrio-podrian-desaparecer-en-colombia-segun-fenalco/202119/

[3] K. Palacios, "Colombia: Lo que sucede con las tiendas de barrio en la pandemia", america-retail, 2020. [Online]. Available at: https://www.america-retail.com/colombia/colombia-lo-que-sucede-con-las-tiendas-de-barrio-en-la-pandemia/

[4] "La transformación digital de las Pymes llegó para quedarse: 8 de cada 10 continuarán con el proceso de reinvención de su objetivo de negocio después de la pandemia.", News Center Latinoamérica, 2021. [Online]. Available at: https://news.microsoft.com/es-xl/la-transformacion-digital-de-las-pymes-llego-para-quedarse-8-de-cada-10-continuaran-con-el-proceso-de-reinvencion-de-su-objetivo-de-negocio-despues-de-la-pandemia/

[5] R. Hernandez, "Tiendas de barrio, una fuente de abastecimiento durante la cuarentena", Radionacional.co, 2020. [Online]. Available at: https://www.radionacional.co/actualidad/tiendas-de-barrio-una-fuente-de-abastecimiento-durante-la-cuarentena#:~:text=Seg%C3%BAn%20un%20estudio%20realizado%20por,ciento%20son%20cabeza%20de%20hogar

[6] "Surtiapp", Tienda.surtiapp.com.co. [Online]. Available at: https://tienda.surtiapp.com.co/

[7] Chiper. [Online]. Available at: https://chiper.co/

[8] J. Ureña et al., "Informe Estudio Nacional de Emprendimiento a Tenderos, primera ronda", Urosario.edu.co, 2020. [Online]. Available at: https://www.urosario.edu.co/PortalUrosario/media/Universidad-del-Rosario-V3/Investigaci%C3%B3n/UCD%202020/Informe-Estudio-Nacional-de-Emprendimiento-a-Tenderos.pdf

[9] I. Bernal, "La radiografía de las tiendas de barrio en el país muestra que más de 52 % son administradas por mujeres", Diario La República, 2022. [Online]. Available at: https://www.larepublica.co/empresas/la-radiografia-de-las-tiendas-de-barrio-mas-de-52-son-administradas-por-mujeres-3433552

[10] "Dígame qué edad tiene y le diré cúal app usa", Asociación Colombiana de Empresas de Investigación de Mercados y Opinión Pública, 2018. [Online]. Available at: https://acei.co/digame-que-edad-tiene-y-le-dire-cual-app-usa/

[11] Twilio. [Online]. Available at: https://www.twilio.com/

[12] Cliengo. [Online]. Available at: https://www.cliengo.com/

[13] "Boletín trimestral del sector TIC - Cifras primer trimestre de 2022", ColombiaTIC, 2022. [Online]. Available at: https://colombiatic.mintic.gov.co/679/articles-238235_archivo_pdf.pdf

[14] C. Alvino, "Estadísticas de la situación digital de Colombia en el 2020-2021", Branch Agencia, 2021. [Online]. Available at: https://branch.com.co/marketing-digital/estadisticas-de-la-situacion-digital-de-colombia-en-el-2020-2021/

[15] "IBM Watson Assistant - AI Chatbot", Ibm.com. [Online]. Available at: https://www.ibm.com/products/watson-assistant/artificial-intelligence

[16] B. Jassova, "What Are the Top Benefits of Chatbots in Business?", landbot.io, 2022. [Online]. Available at: https://landbot.io/blog/benefits-of-chatbots

[17] "MAKING IT PERSONAL", Accenture.com, 2018. [Online]. Available at: https://www.accenture.com/_acnmedia/PDF-77/Accenture-Pulse-Survey.pdf

[18] "New Epsilon research indicates 80 % of consumers are more likely to make a purchase when brands offer personalized experiences", Epsilon.com, 2018. [Online]. Available at: https://www.epsilon.com/us/about-us/pressroom/new-epsilon-research-indicates-80-of-consumers-are-more-likely-to-make-a-purchase-when-brands-offer-personalized-experiences

[19] B. Jassova, "Chatbot Statistics 2021: State of the Market & Opportunities", landbot.io, 2021. [Online]. Available at: https://landbot.io/blog/chatbot-statistics-compilation

[20] P. Vargas, "Conozca algunas de las ventajas de las empresas que tienen chatbots", Diario La República, 2018. [Online]. Available at: https://www.larepublica.co/finanzas-personales/conozca-algunas-de-las-ventajas-de-las-empresas-que-tienen-chatbots-2764214

[21] S. Smith, "Bank Cost Savings via Chatbots to Reach $7.3 Billion by 2023, as Automated Customer Experience Evolv", Juniperresearch.com. [Online]. Available at: https://www.juniperresearch.com/press/bank-cost-savings-via-chatbots-reach-7-3bn-2023

[22] S. Kemp, "Informe digital 2022: las nuevas estadísticas de redes sociales", Social Media Marketing & Management Dashboard, 2022. [Online]. Available at: https://blog.hootsuite.com/es/informe-digital-estadisticas-de-redes-sociales/#El_numero_de_usuarios_de_Telegram_esta_por_los_cielos

[23] A. Navarro, J. Fernández and J. Morales, Redalyc.org, 2013. [Online]. Available at: https://www.redalyc.org/pdf/4962/496250736004.pdf

[24] Blog de Salesforce. 2021. Qué son las metodologías ágiles y cómo pueden ayudar. [online] Available at: https://www.salesforce.com/mx/blog/2021/12/que-son-metodologias-agiles-y-como-pueden-ayudar-a-tus-equipos-de-trabajo.html

[25] "What is Scrum?", Scrum.org, 2020. [Online]. Available at: https://www.scrum.org/resources/what-is-scrum

[26] "What Is Scrum Methodology? & Scrum Project Management", Digite. [Online]. Available at: https://www.digite.com/agile/scrum-methodology/

[27] D. Gonzalez, "Aplicación de técnicas de Design Thinking y metodologías ágiles en procesos de investigación cualitativa. -Casos con tesis doctorales.", research gate, 2017. [Online]. Available at: https://www.researchgate.net/profile/Dora-Gonzalez-Banales/publication/322315570_Aplicacion_de_tecnicas_de_Design_Thinking_y_metodologias_agiles_en_procesos_de_investigacion_cualitativa_-Casos_con_tesis_doctorales/links/5b1eaa420f7e9b0e373db90a/Aplicacion-de-tecnicas-de-Design-Thinking-y-metodologias-agiles-en-procesos-de-investigacion-cualitativa-Casos-con-tesis-doctorales.pdf

[28] D. Arenzana, "Principios de usabilidad web de Jacob Nielsen y el diseño UX," Semrush Blog. [Online]. Available at: https://es.semrush.com/blog/usabilidad-web-principios-jakob-nielsen/

[29] W. Sanchez, "La usabilidad en Ingeniería de Software: definición y características," Repositorio Digital de Ciencia y Cultura de El Salvador REDICCES. [Online]. Available at: http://www.redicces.org.sv/jspui/bitstream/10972/1937/1/2.%20La%20usabilidad%20en%20Ingenieria%20de%20Software-%20definicion%20y%20caracteristicas.pdf

[30] Uaeh.edu.mx, 2020. [Online]. Available at: https://www.uaeh.edu.mx/docencia/P_Presentaciones/icea/asignatura/administracion/2020/tecnica-recoleccion-informacion.pdf

[31] D. Forero, "Qué Es la arquitectura de software: Más Allá de la Programación," Platzi. [Online]. Available at: https://platzi.com/blog/que-es-arquitectura-de-software/

[32] I. Morales, "¿Qué son los patrones de diseño? Conoce para qué te sirven," Platzi. [Online]. Available at: https://platzi.com/blog/patrones-de-diseno/

[33] "¿Qué es una API? - Guía sobre las API para principiantes - AWS", Amazon Web Services, Inc.. [Online]. Available at: https://aws.amazon.com/es/what-is/api/

[34] Oracle.com. n.d. ¿Qué es un bot conversacional?. [online] Available at: https://www.oracle.com/co/chatbots/what-is-a-chatbot

[35] RedHat, ¿Qué es un webhook y para qué sirve? . [Online]. Available at: https://www.redhat.com/es/topics/automation/what-is-a-webhook

[36] sproutsocial. n.d. Comercio conversacional. [online] Available at: https://sproutsocial.com/es/glossary/conversational-commerce

[37] React.js. [Online]. Available at: https://es.reactjs.org/

[38] "Welcome to Flask — Flask Documentation (2.2.x)", Flask.palletsprojects.com. [Online]. Available at: https://flask.palletsprojects.com/en/2.2.x/

[39] "Telegram – a new era of messaging", Telegram. [Online]. Available at: https://telegram.org/

[40] "MVC architecture - detailed explanation," InterviewBit. [Online]. Available at: https://www.interviewbit.com/blog/mvc-architecture/

[41] "Balsamiq wireframes - industry standard low-fidelity wireframing software: Balsamiq," Wireframes - Industry Standard Low-Fidelity Wireframing Software. [Online]. Available at: https://balsamiq.com/wireframes/

[42] React. [Online]. Available at: https://react.dev/

[43] "11 reasons why react.js is still popular in 2023?: Ifour technolab," iFour Blog — Custom Software Development USA — Offshore Software Development - ifourtechnolab.com. [Online]. Available at: https://www.ifourtechnolab.com/blog/11-reasons-why-react-js-is-still-popular-in-2023

[44] M. Acibeiro, "¿Qué es el cross-site scripting (XSS) Y cómo puedes evitarlo?," GoDaddy, 10-Oct-2019. [Online]. Available at: https://es.godaddy.com/blog/que-es-el-cross-site-scripting-xss-y-como-puedes-evitarlo/

[45] "Will reactjs still be a relevant front-end tool in 2023?," Will ReactJS still be a relevant front-end tool in 2023?, 05-Jan-2023. [Online]. Available at: https://fireup.pro/blog/will-reactjs-still-be-a-relevant-front-end-tool-in-2023

[46] "Comparison of the usage statistics of react vs. Vue.js vs. angular for websites," W3Techs. [Online]. Available at: https://w3techs.com/technologies/comparison/js-angularjs,js-react,js-vuejs

[47] "Stack overflow developer survey 2022," Stack Overflow, May-2022. [Online]. Available at: https://survey.stackoverflow.co/2022/#most-loved-dreaded-and-wanted-webframe-want

[48] Telegram, "Web apps for bots," Telegram APIs. [Online]. Available at: https://core.telegram.org/bots/webapps

[49] Telegram, "Bot API library examples," Telegram APIs. [Online]. Available at: https://core.telegram.org/bots/samples

[50] "Pytelegrambotapi," PyPI, 18-May-2023. [Online]. Available at: https://pypi.org/project/pyTelegramBotAPI/

[51] M. Frackiewicz, "The advantages of using python for developing Chatbots and conversational AI," TS2 SPACE, 05-Apr-2023. [Online]. Available at: https://ts2.space/en/the-advantages-of-using-python-for-developing-chatbots-and-conversational-ai/

[52] A. Jalli, "Is it worth learning python? 8 reasons to learn [in 2023]," codingem.com, 13-Dec-2022. [Online]. Available at: https://www.codingem.com/is-it-worth-learning-python/

[53] "Python vs. other programming languages," stxnext. [Online]. Available at: https://www.stxnext.com/python-vs-other-programming-languages/

[54] M. Deery, "What Is Flask and How Do Developers Use It? A Quick Guide," CareerFoundry, 01-Aug-2022. [Online]. Available at: https://careerfoundry.com/en/blog/web-development/what-is-flask/#advantages-and-disadvantages-of-flask

[55] Oracle, "¿Qué es una base de datos relacional?," Qué es una base de datos relacional — Oracle Colombia. [Online]. Available: https://www.oracle.com/co/database/what-is-a-relational-database/

[56] J. Ellingwood, "PostgreSQL advantages: Benefits of using PostgreSQL," Prisma's Data Guide. [Online]. Available at: https://www.prisma.io/dataguide/postgresql/benefits-of-postgresql#robust-feature-set

[57] "Stack overflow developer survey 2022," Stack Overflow. [Online]. Available at: https://survey.stackoverflow.co/2022/#most-popular-technologies-database-prof

[58] "Dialogflow Documentation — Google Cloud", Google Cloud. [Online]. Available at: https://www.dialogflow.com

[59] "Security-first diagramming for teams.," draw.io. [Online]. Available at: https://www.drawio.com/

[60] "Sqlalchemy," SQLAlchemy. [Online]. Available at: https://www.sqlalchemy.org/

[61] "Amazon Relational Database Service," Amazon RDS. [Online]. Available at: https://aws.amazon.com/es/rds/

[62] "Automatically import your data into Google Sheets," Coefficient, 27-Apr-2023. [Online]. Available at: https://coefficient.io/

[63] "Develop and deploy websites and apps in record time," Netlify. [Online]. Available at: https://www.netlify.com/

[64] R. Shtylman, "Localtunnel expose yourself to the world," Localtunnel Expose yourself to the world. [Online]. Available at: https://localtunnel.me/

[65] ngrok. [Online]. Available at: https://ngrok.com/

[66] Universidad de Champagnat - Licenciatura en RR.HH., "Encuesta, Tipos y procedimiento de uso en Investigación de Mercados," gestiopolis, 01-Aug-2019. [Online]. Available at: https://www.gestiopolis.com/encuesta-tipos-y-procedimiento-de-uso-en-investigacion-de-mercados/

Capítulo 10

Anexos

10.1. Requisitos

10.1.1. Requisitos Funcionales

A continuación, se visualizarán los requisitos funcionales que se han considerado para el desarrollo del Chatbot.

Identificación	Nombre	Descripción	Tipo	Estado	Versión	Prioridad	Estado	Ultima fecha estado
RF-01	Integración con plataforma de chat	El chatbot debe poder integrarse con alguna plataforma de chat relevante, como WhatsApp o Telegram.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-02	Integración con sistemas externos	El chatbot debe poder integrarse con sistemas externos, como lo son las bases de datos, para proporcionar respuestas más detalladas y específicas.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-03	Respuestas automáticas	El chatbot debe poder proporcionar respuestas automáticas a preguntas comunes, como horarios de atención al cliente e información de contacto.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-04	Capacidad de seguimiento	El chatbot debe ser capaz de seguir la conversación y recordar la información relevante, como los detalles de un pedido o los datos del cliente.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-05	Capacidad de análisis	El chatbot debe ser capaz de recopilar y analizar datos para ayudar a mejorar la eficacia y experiencia del usuario.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-06	Personalización	El chatbot debe poder personalizarse para adaptarse a las necesidades específicas del negocio, como la marca y los productos que se venden.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-07	Interfaz del usuario	El chatbot debe poseer una interfaz que permita un alto grado de acoplamiento entre el usuario y el sistema.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-08	Datos del Cliente.	El chatbot debe contar con datos del cliente tales como: Identificación, nombres, apellidos, nombre del negocio, dirección, correo electrónico, celular, barrio, ciudad y fecha de creación o registro.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-09	Datos de los Productos.	El chatbot debe contar con datos del producto tales como: Código, Categoría, Descripción, Precio, Cantidad e Imagen.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-10	Datos del pedido.	El chatbot debe contar con datos del pedido tales como: Identificación del cliente que realiza el pedido, nombres, apellidos, nombre del negocio, direcoción, correo electrónico, celular, barrio, ciudad, fecha del pedido, fecha máxima de entrega y total a pagar.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023

Figura 10.1: Requisitos Funcionales del. 1-10.

Identificación	Nombre	Descripción	Tipo	Estado	Versión	Prioridad	Estado	Ultima fecha estado
RF-11	Registro del Cliente	El chatbot debe permitirle al cliente poder registrarse ingresando los datos correspondientes al RF-08 mediante campos de texto.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-12	Integración con el sistema de inventario	El chatbot debe estar integrado con el sistema de inventario para mantener la información de los productos actualizada y asegurarse de que los productos estén disponibles para su compra.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-13	Validación de Identificación	El chatbot debe permitirle al cliente ingresar un número de identificación para validar en base de datos la autenticidad o existencia del mismo.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-14	Proceso del pedido	El chatbot debe poder guiar al cliente a través del proceso del pedido, incluyendo agregar productos al carrito, realizar pedido y proporcionar información del mismo.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-15	Catálogo	El cliente podrá visualizar en la WebApp del chatbot los distintos productos que se pueden añadir al carrito de compras.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-16	Categorías	El cliente podrá visualizar en la WebApp del chatbot las distintas categorías de los productos.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-17	Carrito de Compras	El Carrito de compras debe permitir al cliente visualizar el listado de productos que se está pidiendo, su precio , cantidad y de igual manera, poder modificar la cantidad de los productos.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-18	Búsqueda de productos	El chatbot debe poder ayudar al cliente a buscar y encontrar productos en el catálogo en línea.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-19	Menú de Búsqueda	El chatbot desde la WebApp debe permitirle al cliente poder filtrar productos del catálogo dependiendo su categoría por medio de un menú desplegable.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-20	Ayuda	El chatbot debe contar con una sección de ayuda en la cual se le explicará al usuario por medio de un video como se puede registrar y/o realizar un pedido	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023

Figura 10.2: Requisitos Funcionales del. 11-20.

Identificación	Nombre	Descripción	Tipo	Estado	Versión	Prioridad	Estado	Ultima fecha estado
RF-21	Salir	El chatbot debe contar con la opción "Salir" para poder dar por acabada la conversación con el cliente.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-22	Visualizar Información de Contacto	El chatbot debe permitirle al cliente poder acceder a la información de la empresa o proveedor como lo son el teléfono, celular, correo o dirección para poder ponerse en contacto directamente con ellos.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-23	Validar cantidad en inventario	El chatbot debe poder validar la cantidad que hay en el inventario de X producto, para que en caso tal de que el producto no se encuentre disponible, se notifique tanto al cliente como al proveedor que este no se encuentra disponible y que de igual manera no afecte el total a pagar del pedido.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-24	Aviso cantidad en inventario	El chatbot debe poder notificar al proveedor cuando algún producto se esté escaseando, para que este pueda suplir la cantidad que hay en el inventario.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-25	Realizar Pedido	El chatbot después de haber verificado la existencia de un cliente en la base de datos debe permitir al mismo empezar el proceso para realizar un pedido al proveedor.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-26	Visualizar WebApp	El chatbot debe permitir al cliente abrir la aplicación web dentro de la conversación con el bot al presionar un botón.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-27	Visualizar Información de Pedido - Cliente	El chatbot debe enviar un mensaje al cliente listando los productos pedidos y el valor total a pagar por parte del cliente junto con sus datos de identificación.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-28	Visualizar Información de Pedido - Proveedor	El chatbot debe enviar un mensaje al proveedor listando los productos pedidos y el valor total a pagar por parte del cliente.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-29	Integración con excel	El chatbot debe poder integrar las bases de datos con excel para poder facilitar al proveedor modificar el inventario y así mismo, poder importar y exportar datos.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RF-30	Grupo en plataforma de chat	El chatbot debe poder agregarse a un grupo en la plataforma de chat seleccionada para que de esta manera pueda enviar mensajes al grupo en el cual estarán las personas encargadas de despachar dichos pedidos y su supervisor.	Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023

Figura 10.3: Requisitos Funcionales del. 21-30.

10.1.2. Requisitos No Funcionales

A continuación, se visualizarán los requisitos no funcionales que se han considerado para el desarrollo del Chatbot.

Identificación	Nombre	Descripción	Tipo	Estado	Versión	Prioridad	Estado	Ultima fecha estado
RNF-1	Persistencia de la Información	El chatbot debe proveer la acción de preservar la información de un objeto de forma permanente, pero a su vez también se debe poder recuperar la información del mismo para que pueda ser nuevamente utilizado.	No Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RNF-2	Tiempo de Respuesta	El chatbot debe contar con un tiempo de respuesta no mayor a 5 segundos por realización de funcionalidad.	No Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RNF-3	Disponibilidad	El chatbot debe proveer de alta disponibilidad para así poder garantizar la continuidad de los servicios.	No Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RNF-4	Usabilidad	El chatbot debe ser fácil de usar y comprender.	No Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023
RNF-5	Mantenibilidad	El chatbot debe ser fácil de mantener y actualizar para que sea sostenible en el tiempo.	No Funcional	Por aprobar	1.0	Alta	Por completar	19/04/2023

Figura 10.4: Requisitos No Funcionales del. 1-5.

10.2. Prototipos Previos al Desarrollo

10.2.1. Prototipos de vistas usuario Tendero - Registro

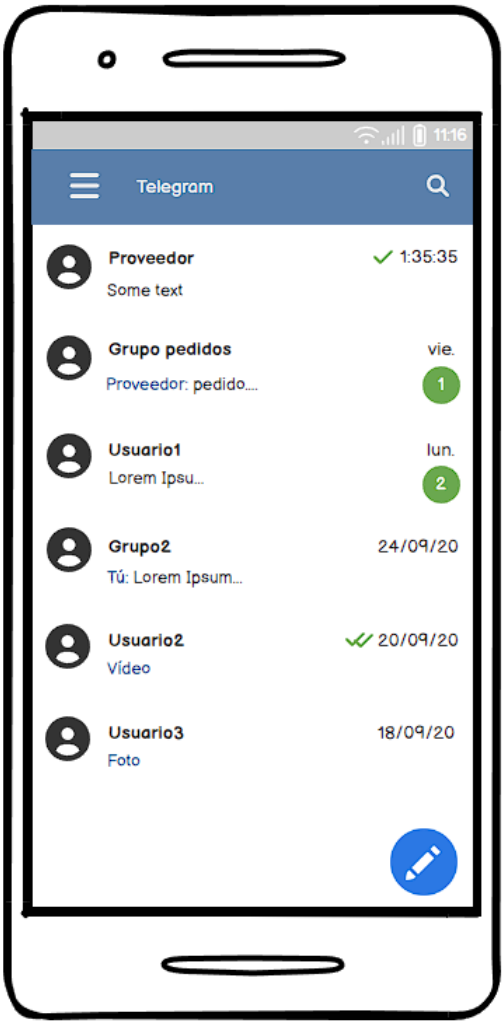


Figura 10.5: Prototipo vista de inicio de Telegram.

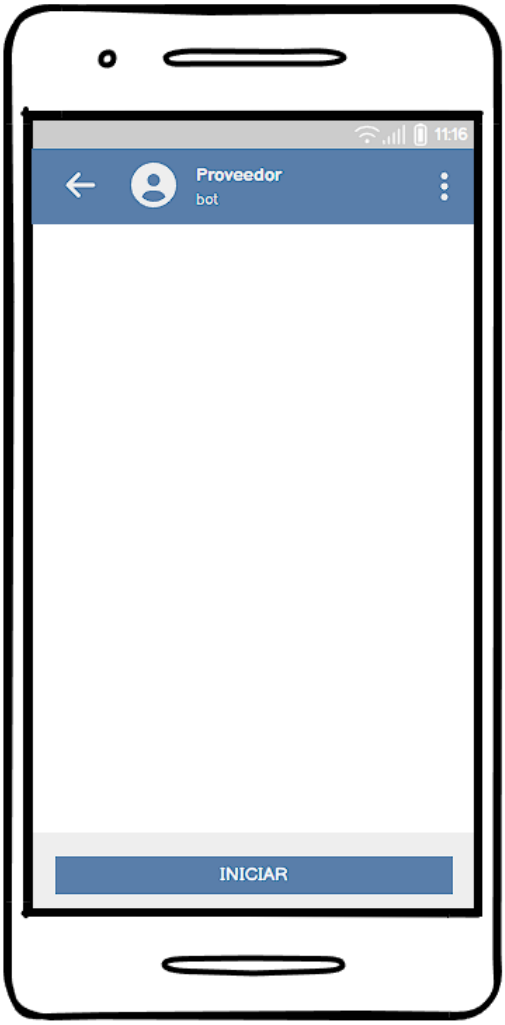


Figura 10.6: Prototipo de inicio de conversación con el Chatbot.

Figura 10.7: Prototipo conversación Chatbot registro primera parte.

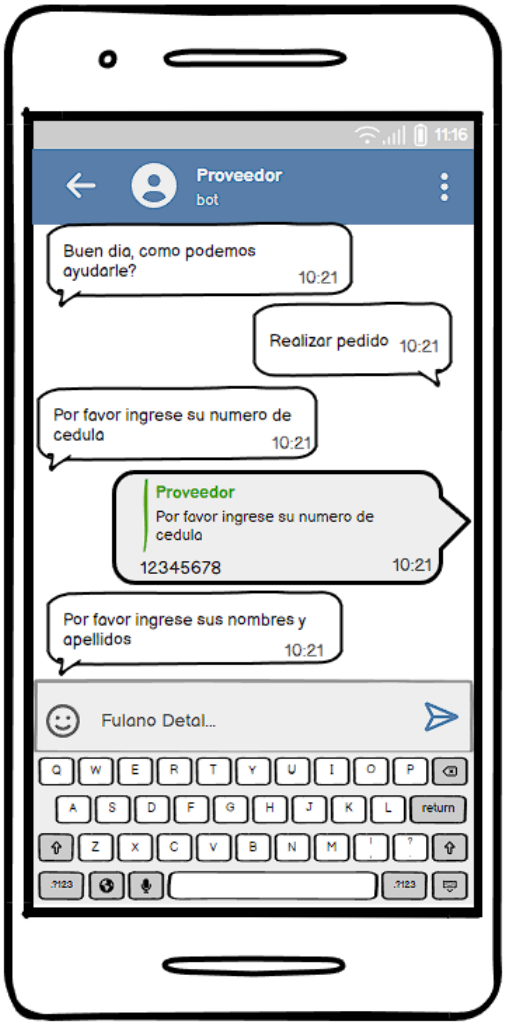


Figura 10.8: Prototipo conversación Chatbot registro segunda parte.

Figura 10.9: Prototipo conversación Chatbot registro finalizado.

10.2.2. Prototipos de vistas usuario Tendero - Realizar pedido

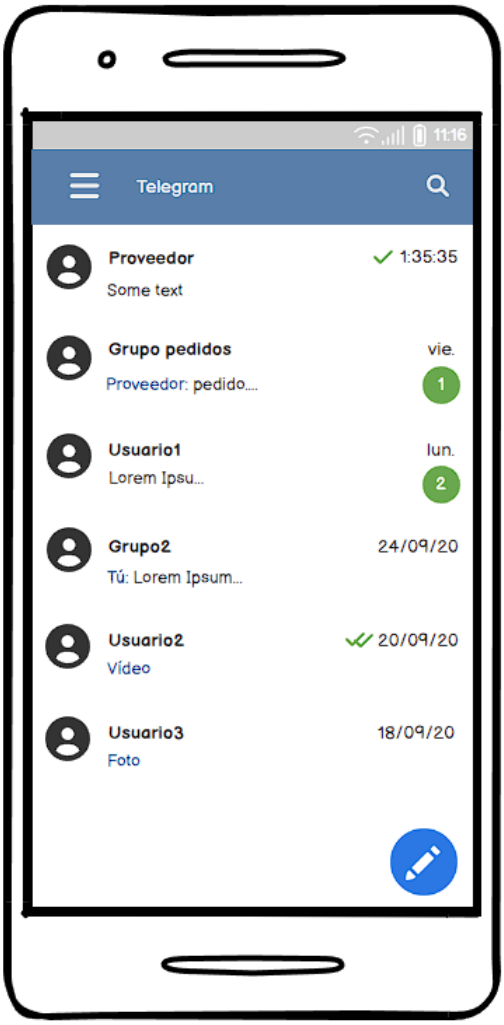


Figura 10.10: Prototipo vista de inicio de Telegram.

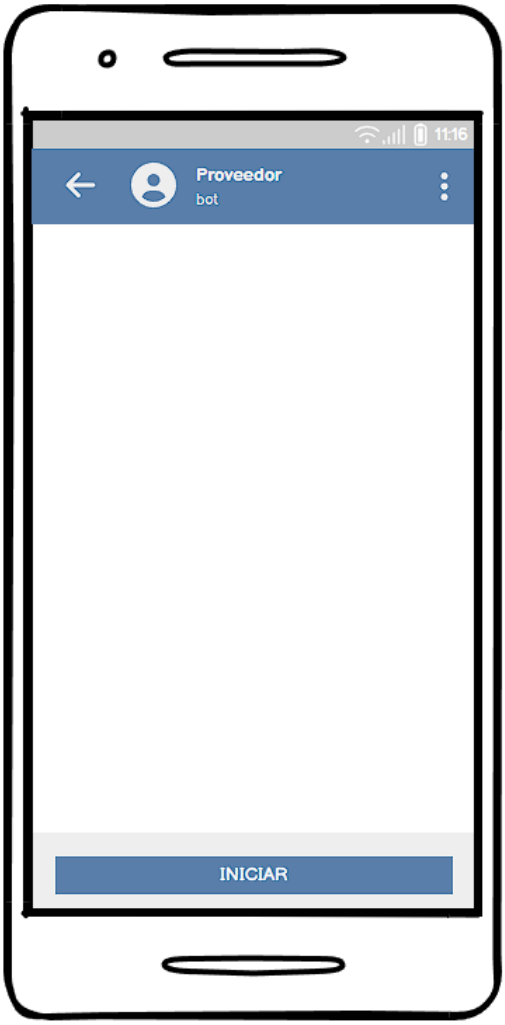


Figura 10.11: Prototipo de inicio de conversación con el Chatbot.

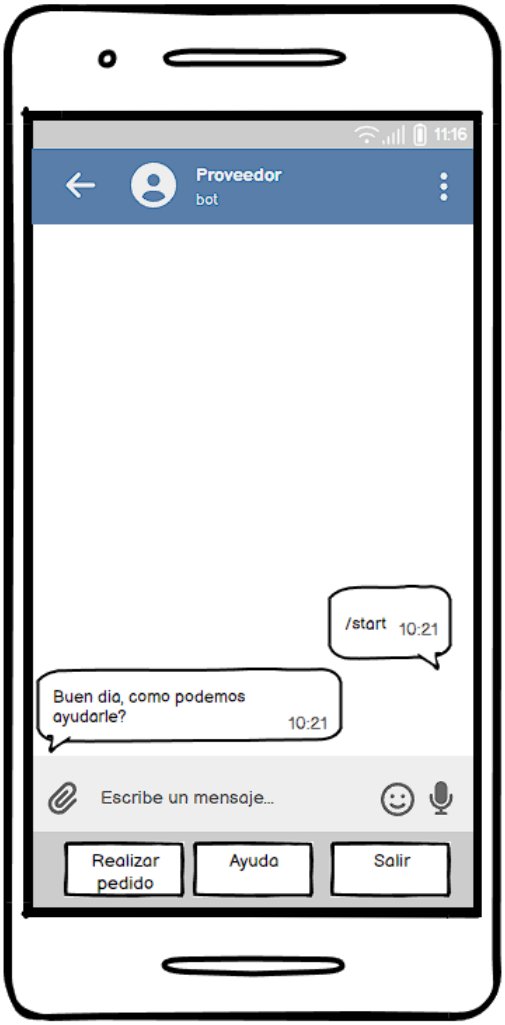


Figura 10.12: Prototipo conversación Chatbot primera parte.

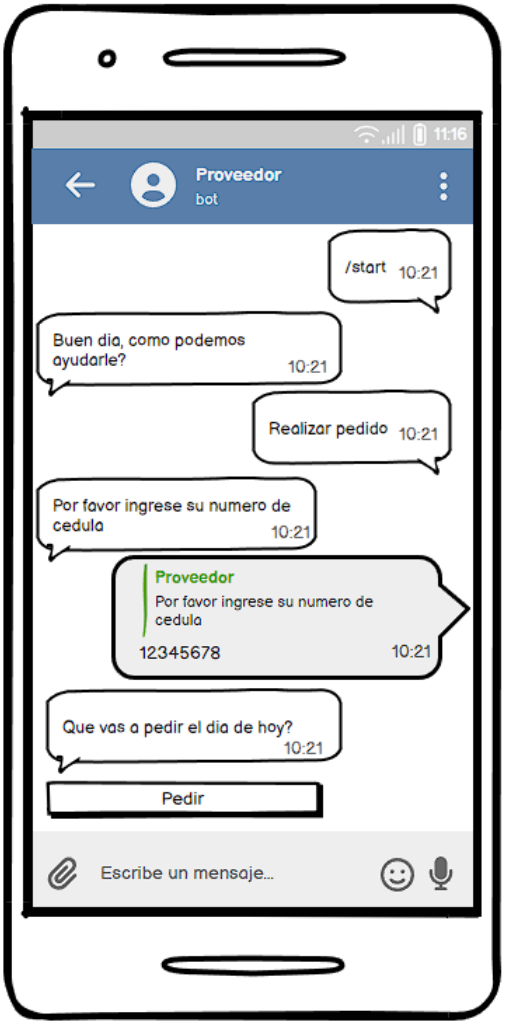


Figura 10.13: Prototipo conversación Chatbot segunda parte.

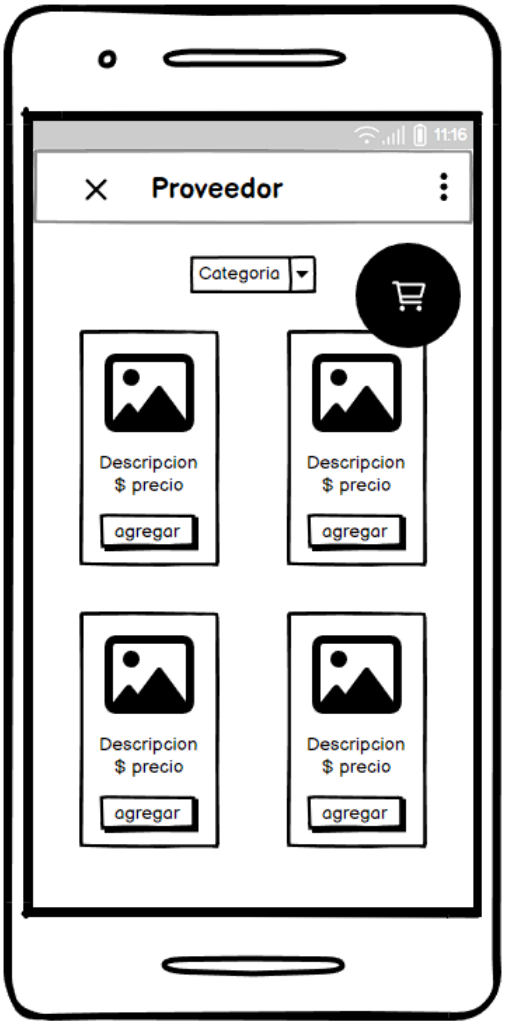


Figura 10.14: Prototipo vista selección productos.

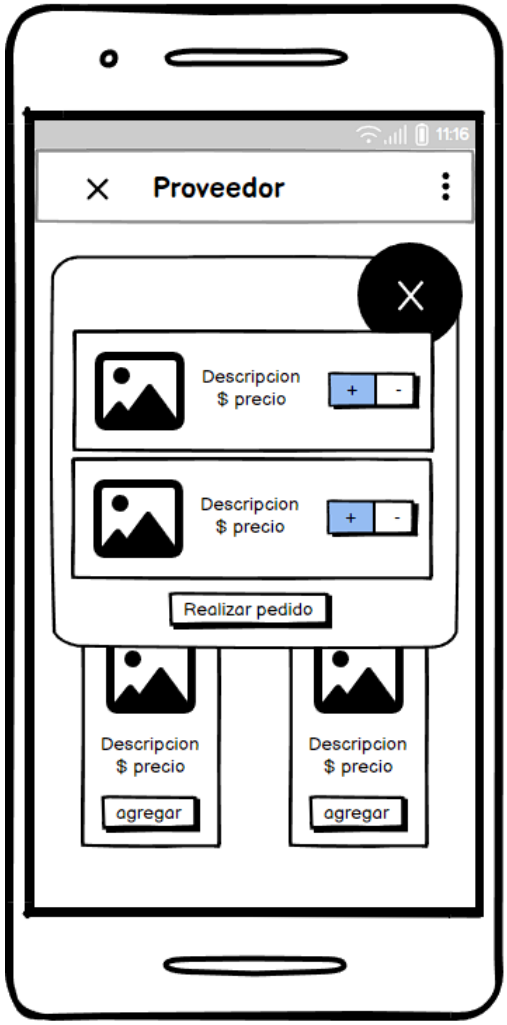


Figura 10.15: Prototipo vista carrito de compras.

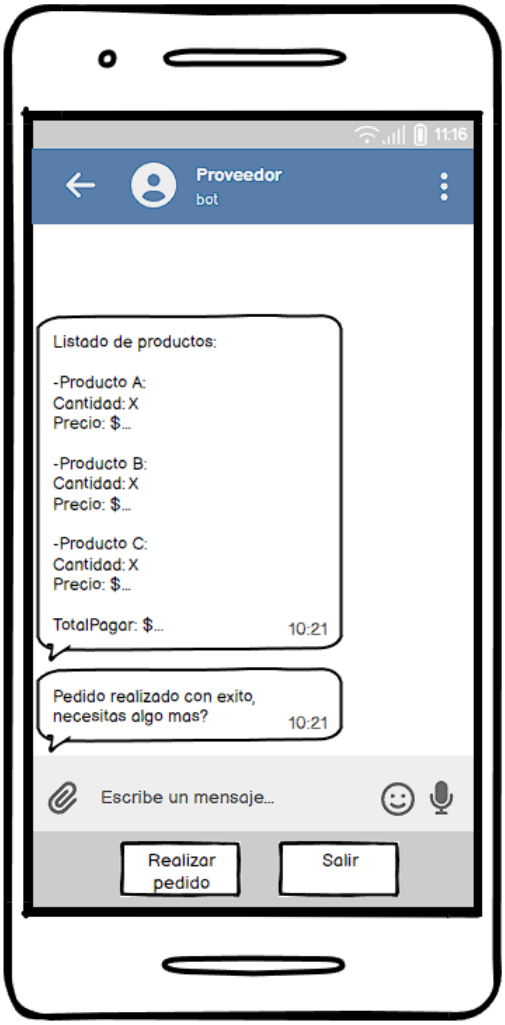


Figura 10.16: Prototipo del pedido realizado con éxito.

10.2.3. Prototipos de vistas usuario Proveedor

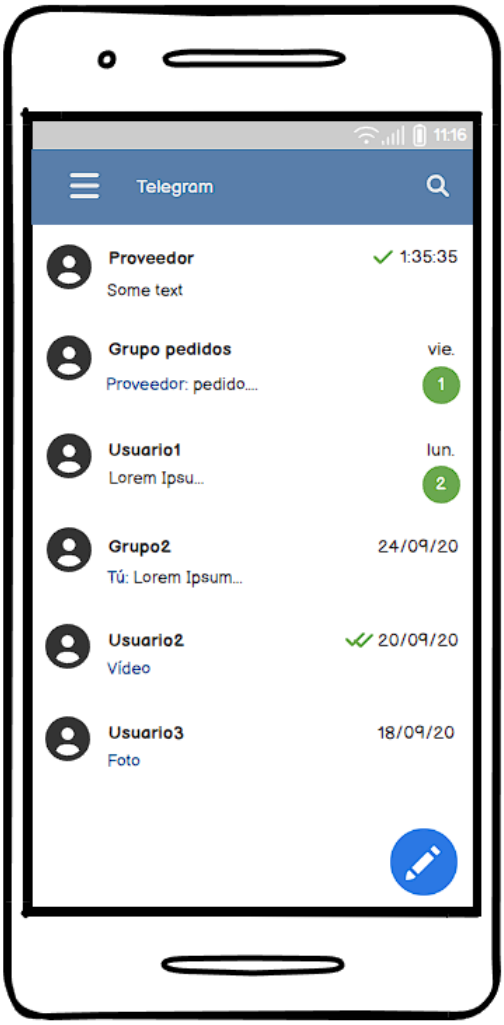


Figura 10.17: Prototipo vista grupo Telegram proveedor.

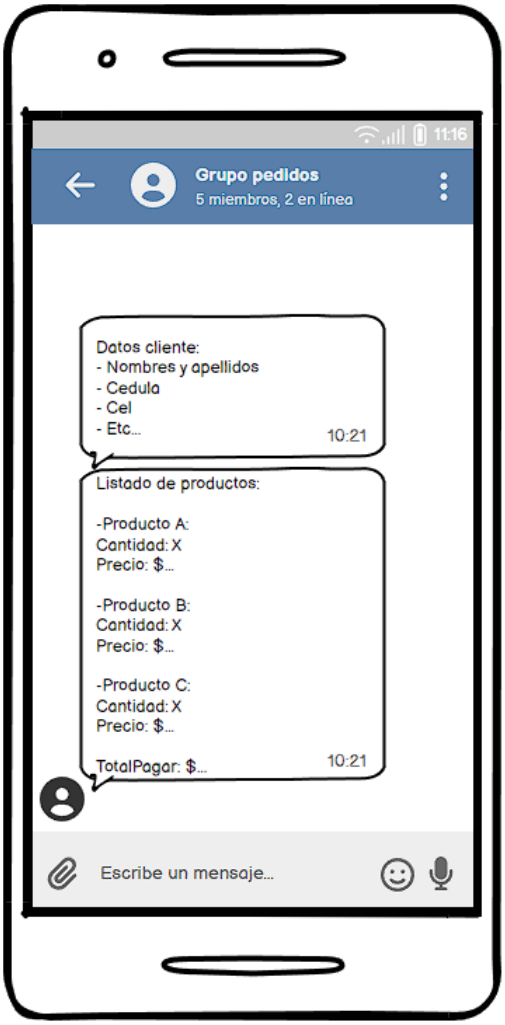


Figura 10.18: Prototipo vista pedidos - grupo Telegram proveedor.

10.3. Prototipo Final Funcional

A continuación, se explicará el proceso que se debe llevar a cabo para poder iniciar de manera exitosa una conversación con el Chatbot en Telegram:

Al abrir la aplicación de Telegram en su dispositivo móvil, se encontrará con la siguiente vista que corresponde al menú de inicio de Telegram.

Figura 10.19: Vista proceso de Inicio Parte I.

En el cual usted puede seleccionar un chat existente para chatear con alguien, comenzar una nueva conversación con alguno de sus contactos al presionar el icono del lápiz, seleccionar el menú en la parte superior de la izquierda o seleccionar la lupa (marcada en el recuadro rojo) en la parte superior derecha para buscar a alguien, que es lo que se va a hacer en este caso para poder buscar el Chatbot del proveedor.

Posteriormente, como se puede apreciar en la siguiente vista, se procede a buscar el Chatbot que en este caso tiene el nombre del proveedor, el cual es Distri Romel y se selecciona para poder iniciar una conversación con el mismo.

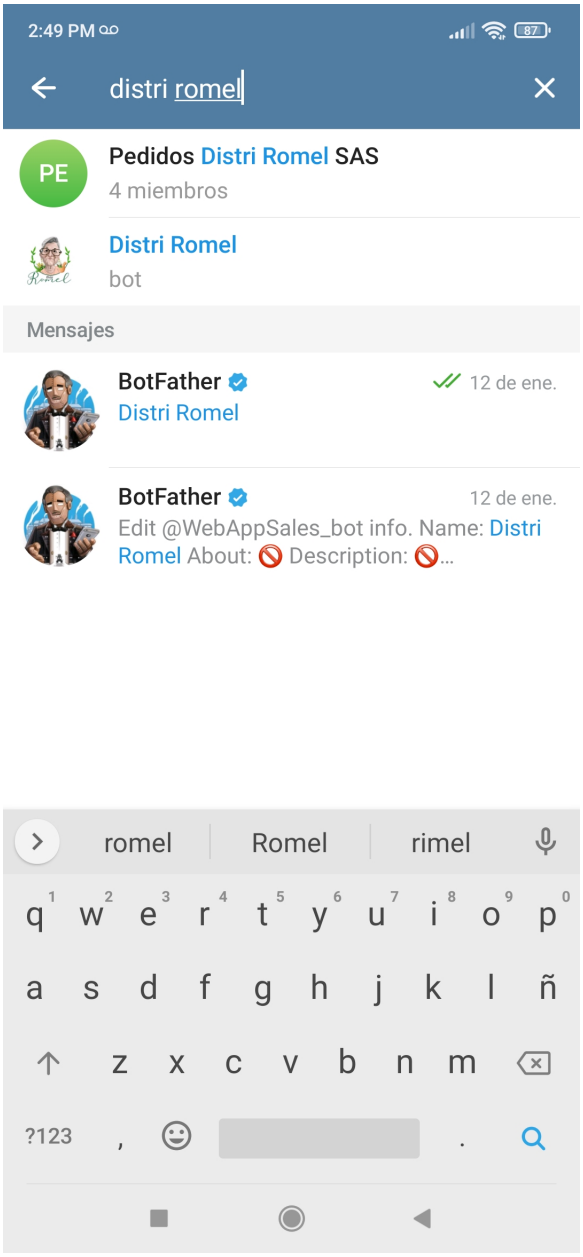


Figura 10.20: Vista proceso de Inicio Parte II.

En esta vista se presionará el botón grande azul de iniciar que aparece en la parte inferior, el cual es el que permite enviar un comando al Chatbot para iniciar una conversación.

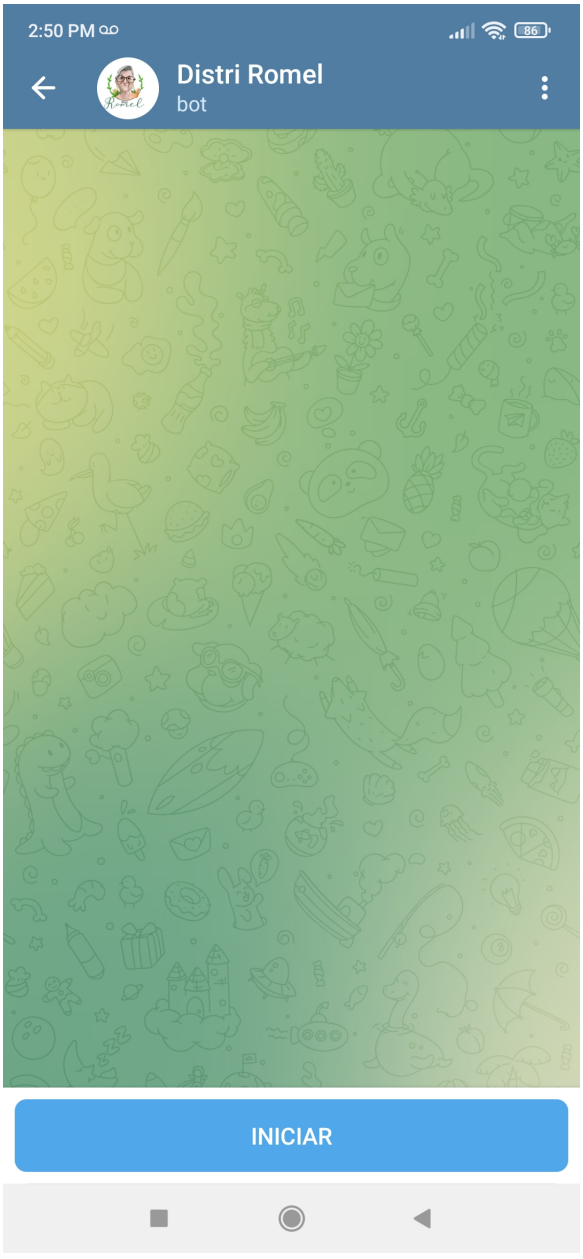


Figura 10.21: Vista proceso de Inicio Parte III.

Seguidamente, luego de haber presionado el botón de inicio, se encontrará con la siguiente vista en la cual se topará con el menú principal del Chatbot; este dispone de los botones “Realizar pedido”, “Contactenos”, “Ayuda” y “Salir”. Botones con los cuales usted podrá interactuar de acuerdo con lo que necesite.

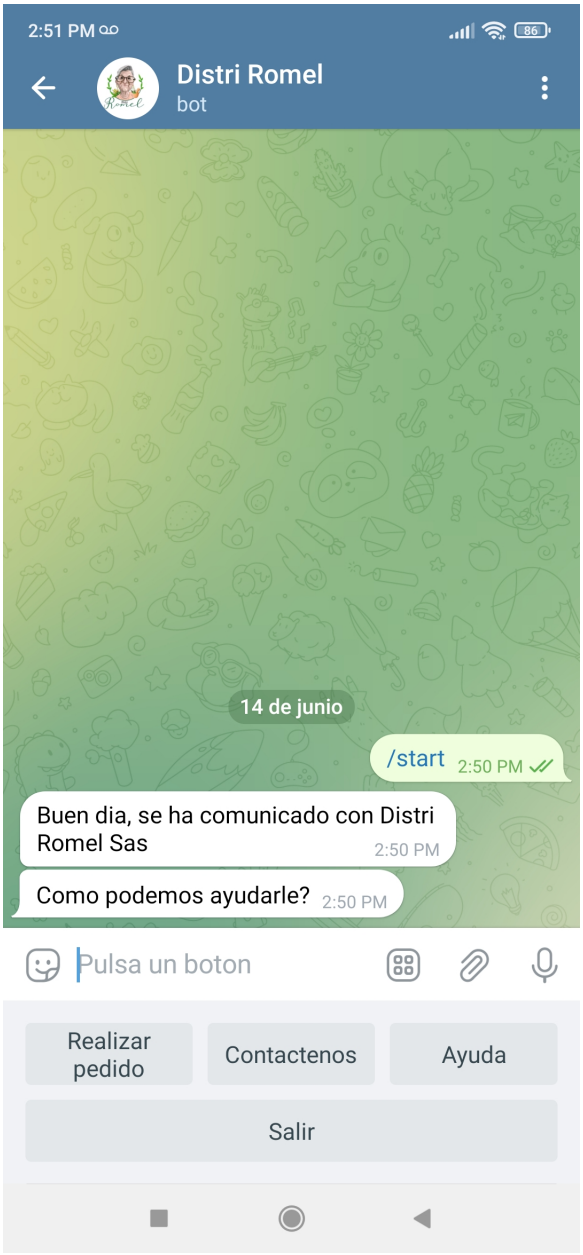


Figura 10.22: Vista proceso de Inicio Parte IV.

En caso tal de que el usuario tendero presione el botón de “Ayuda”, este accederá a la siguiente vista en la cual, como se puede ver, se le enviaran dos enlaces con los que puede acceder a dos videos en Youtube que lo guiaran tanto en el proceso de registro, como en el de la realización de su pedido.

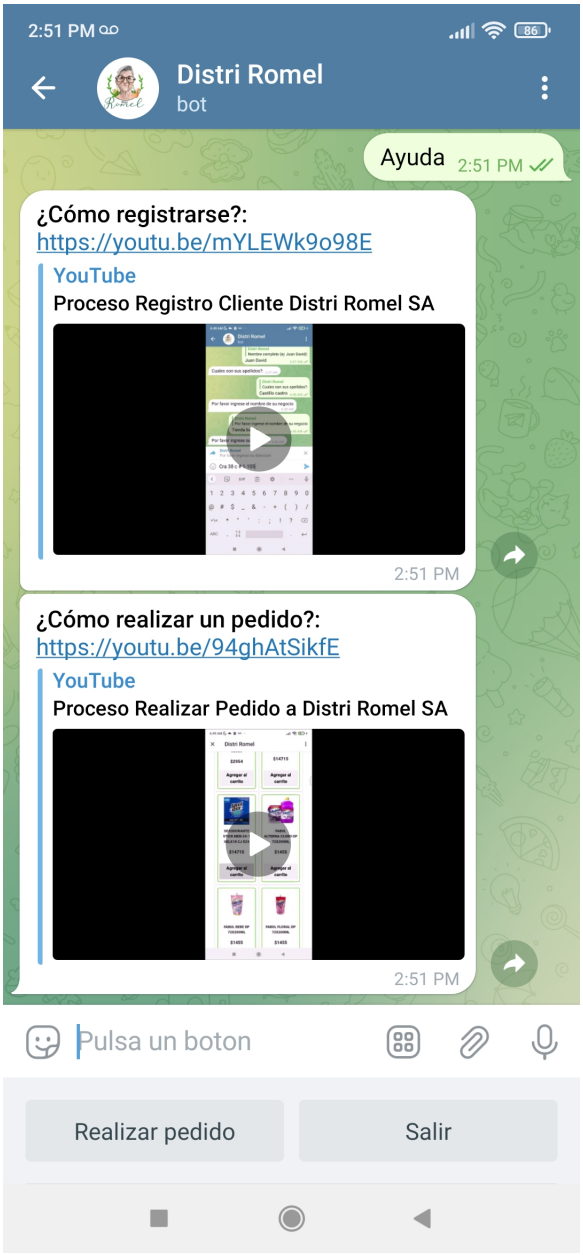


Figura 10.23: Vista sección Ayuda.

Por otra parte, en caso tal de que el usuario tendero presione el botón de “Contactenos”, este accederá a la siguiente vista en la cual, como se puede ver, se le enviaran datos relevantes del Proveedor que le permitirán ponerse en contacto, como lo son sus redes sociales, horarios de atención, teléfonos, dirección y correo.

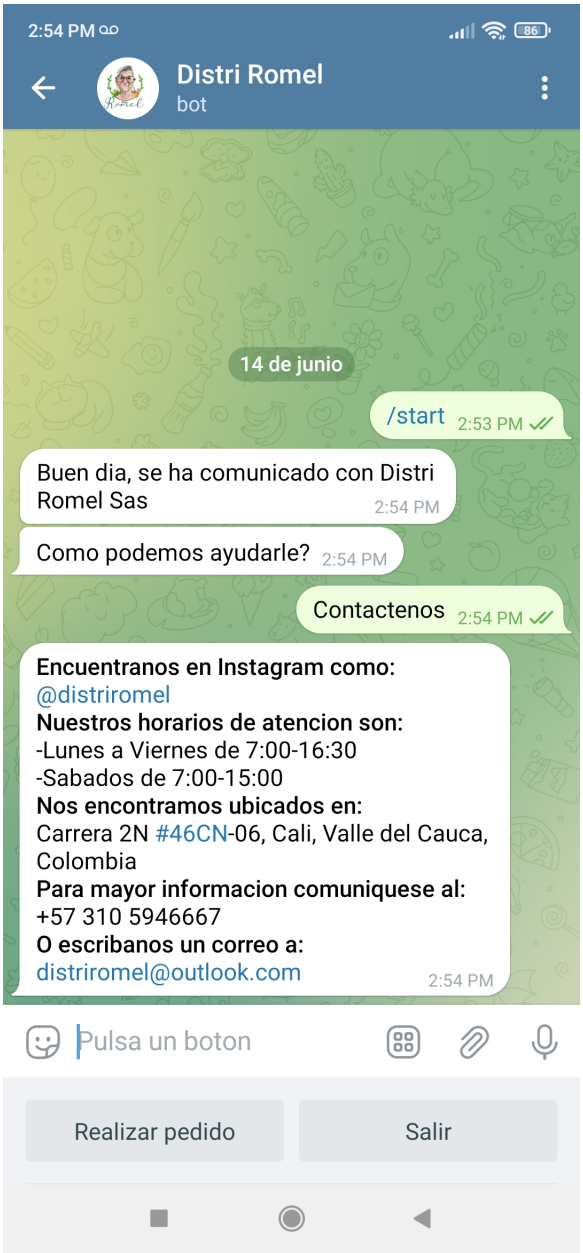


Figura 10.24: Vista sección Contáctenos.

10.3.1. Registro Usuario Tendero

A continuación, se explicará el proceso que se debe llevar a cabo para poder realizar de forma exitosa el registro de un usuario Tendero:

Para poder comenzar el proceso de registro de un Tendero, en el menú principal del Chatbot se debe presionar el botón de "Realizar pedido" como se muestra en la vista siguiente señalada por el recuadro rojo.

Figura 10.25: Vista proceso de Registro Tendero Parte I.

Una vez se haya presionado el botón de “Realizar pedido”, como se puede apreciar en la vista siguiente, se solicitará la cédula del usuario Tendero para verificar si este ya existe en la base de datos del Proveedor.

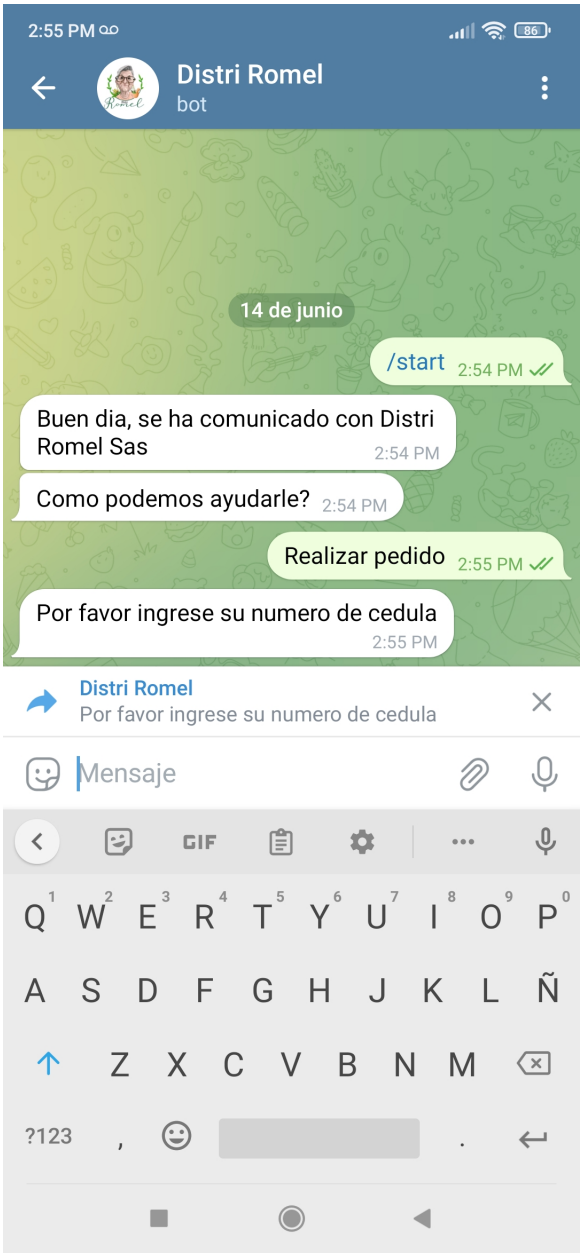


Figura 10.26: Vista proceso de Registro Tendero Parte II.

En este caso, al ser la primera vez que se comunica con el Proveedor y al verificar que no se encuentra en la base de datos después del proceso de verificación; como se muestra en la siguiente vista, el Chatbot procederá a pedirle ciertos datos relevantes para poder realizar un pedido y poder comunicarse con el proveedor como lo son los nombres, apellidos, celular, dirección, etc.

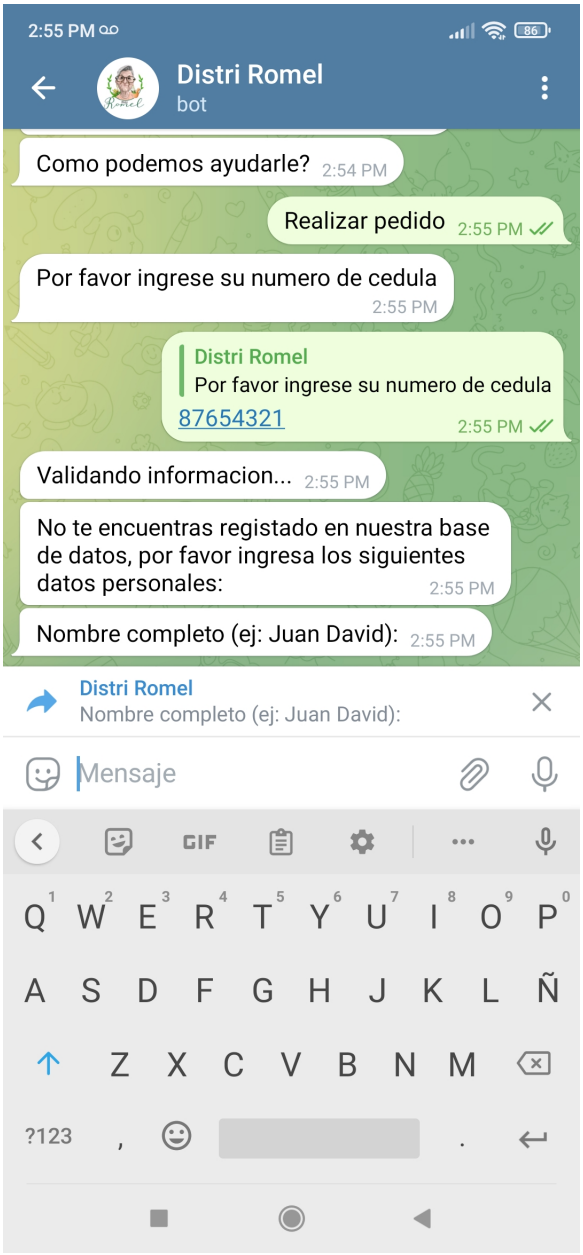


Figura 10.27: Vista proceso de Registro Tendero Parte III.

Una vez se hayan ingresado los datos solicitados por el Chatbot correctamente, se le enviará un mensaje mostrando los datos que usted digitó previamente y con los cuales usted queda registrado oficialmente en la base de datos de su proveedor (en este caso Distri Romel), como se ve en la siguiente vista en la cual después de haber completado este proceso de registro, usted podrá proceder a realizar su pedido.

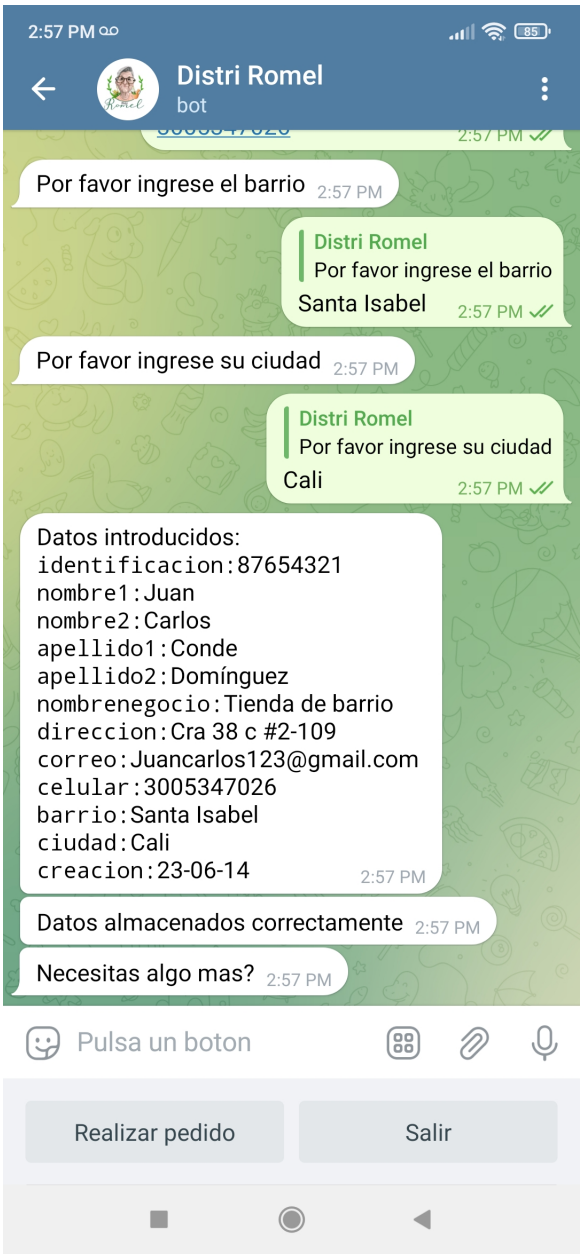


Figura 10.28: Vista proceso de Registro Tendero Parte IV.

10.3.2. Realizar Pedido

A continuación, se explicará el proceso que se debe llevar a cabo para que un usuario Tendero pueda realizar su pedido al Proveedor de manera correcta:

En la vista siguiente, se puede apreciar que una vez que el Usuario Tendero haya completado exitosamente el proceso de registro, en la parte inferior se habilitarán dos botones “Realizar pedido” y “Salir”, con los cuales podrá interactuar, en este caso se selecciona el botón de “Realizar pedido” para poder empezar el proceso de la realización del pedido al Proveedor.

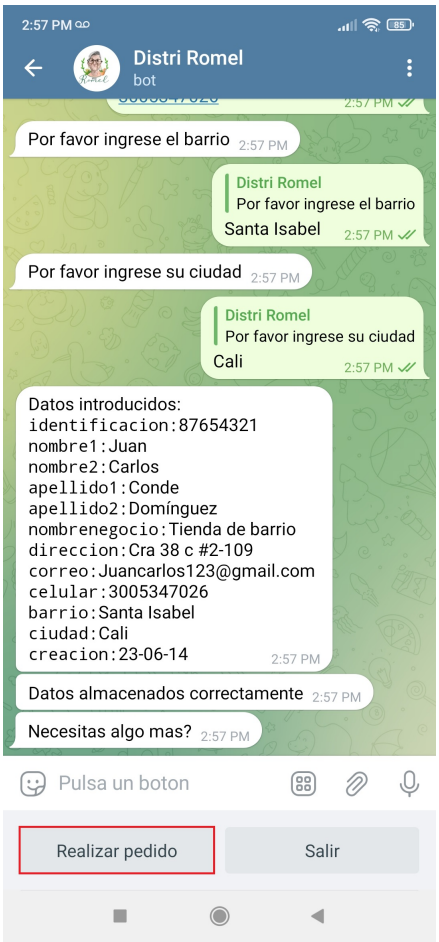


Figura 10.29: Vista proceso Realizar pedido Parte I.

Después de haber presionado el botón de “Realizar pedido”, como se puede ver a continuación, se volverá a pedir el número de identificación del usuario Tendero (la cédula de ciudadanía en este caso), el cual se validará nuevamente que esté registrado en la base de datos del Proveedor, para así una vez validado, darle acceso al botón seleccionado por el recuadro rojo que permitirá abrir la Aplicación Web dentro de la conversación con el Chatbot.

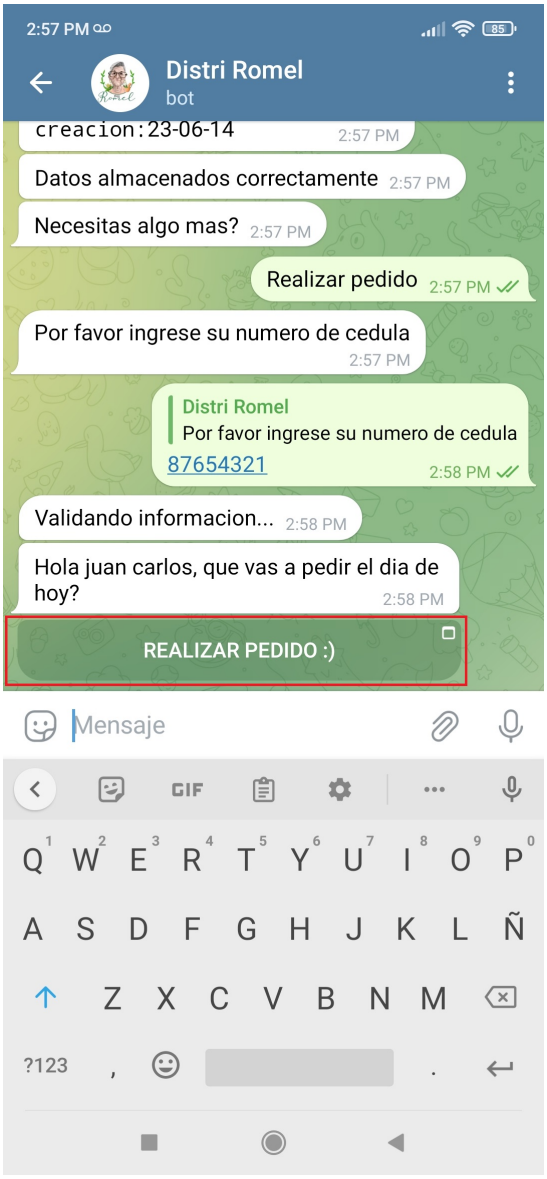


Figura 10.30: Vista proceso Realizar pedido Parte II.

En esta vista, el usuario Tendero podrá visualizar y seleccionar los distintos productos que ofrece el Proveedor, para poder ir agregándolos a un carrito de compras.

Figura 10.31: Vista proceso Realizar pedido Parte III.

Del mismo modo, como se puede apreciar en la siguiente vista, el usuario podrá filtrar los distintos productos por categorías y desplazarse de manera vertical para buscar los productos que requiere.

Figura 10.32: Vista proceso Realizar pedido Parte IV.

Una vez seleccionados los productos por parte del usuario Tendero, este podrá presionar el botón circular negro con el icono de un carro de compras en la parte superior derecha, para poder acceder al carrito de compras, como se ve en la siguiente vista, donde este va a poder modificar su pedido, permitiéndole "agregar" o "sacar" productos del carrito para finalmente una vez este esté satisfecho con su pedido pueda presionar el botón gris de "Realizar pedido" para continuar con el proceso.

Figura 10.33: Vista proceso Realizar pedido Parte V.

Posteriormente, en esta vista siguiente se mostrará una pantalla de carga con el objetivo de mantener al usuario Tendero informado del estado en que se encuentra mientras se está procesando su pedido.

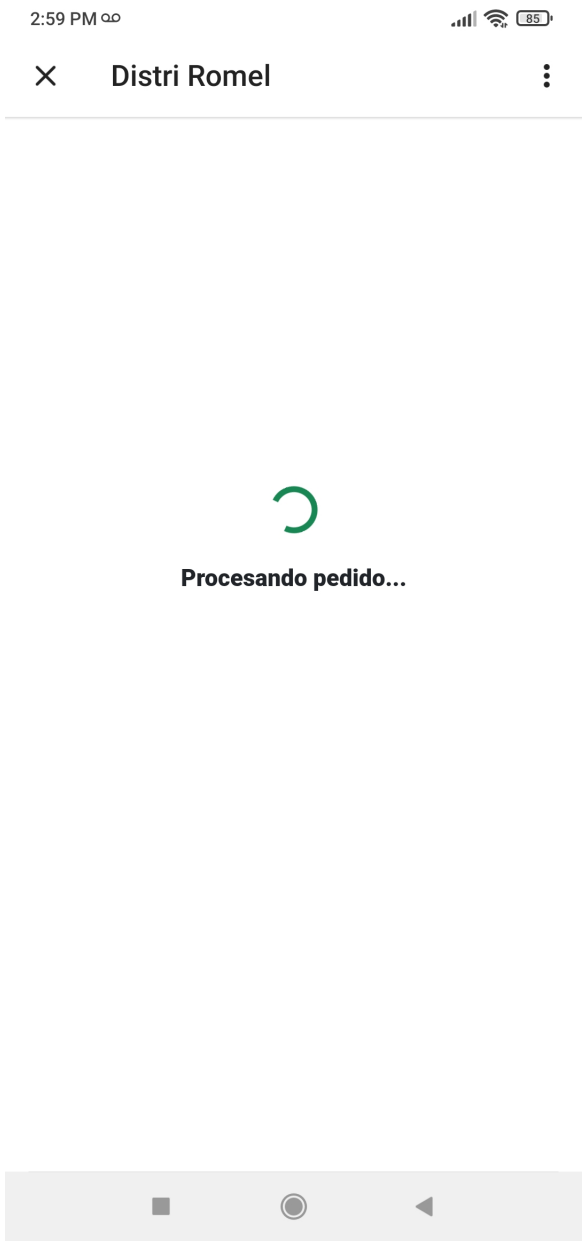


Figura 10.34: Vista proceso Realizar pedido Parte VI.

Finalmente, en esta vista se puede visualizar que se ha completado correctamente el proceso de realización de un pedido, en donde se muestran los datos de la persona registrada previamente, que es la misma que realizó el pedido, la fecha que se realizó el mismo, la fecha máxima de entrega, el total a pagar calculado y también un listado de los productos que se seleccionaron. Estos datos se envían tanto al usuario Tendero como al usuario Proveedor. También se habilitan nuevamente los botones "Realizar pedido", si se quiere hacer algún otro pedido adicional, y el botón "Salir", el cual terminará la conversación con el Chatbot.

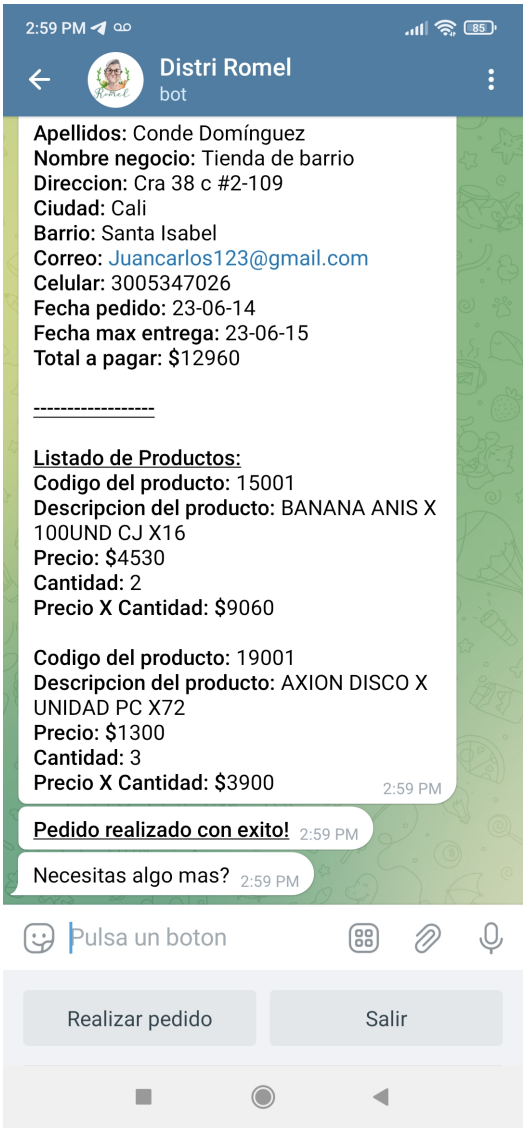


Figura 10.35: Vista proceso Realizar pedido Parte VII.

10.3.3. Visualizar Pedido

A continuación, se explicará el proceso que se debe llevar a cabo para que un usuario Proveedor pueda visualizar los pedidos realizados por los distintos usuarios Tenderos que han interactuado con su Chatbot:

En esta vista, se puede apreciar el menú de inicio de Telegram, en el cual se encuentra el grupo "Pedidos Distri Romel SAS", en el cual no solo están los encargados de distribuir los productos a los tenderos, sino que también está el Chatbot, el cual estará enviando a este grupo todos los pedidos que vaya recibiendo de los Tenderos.

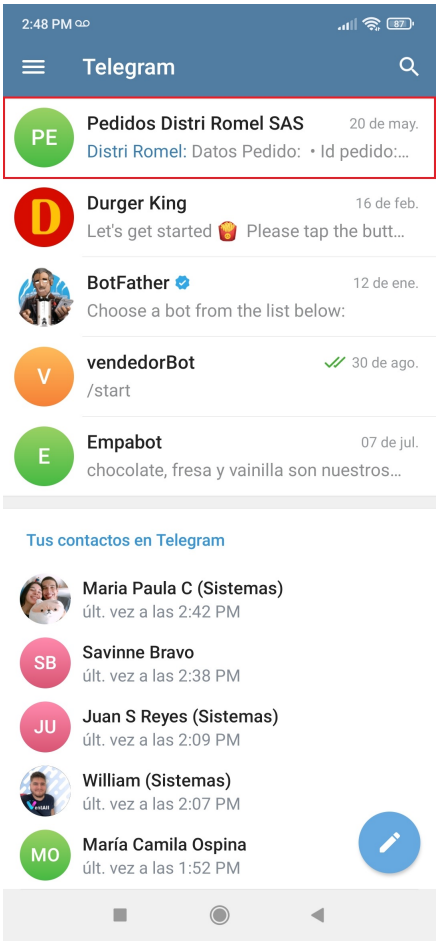


Figura 10.36: Vista pedido recibido Proveedor Parte I.

En esta vista, que corresponde a la conversación del grupo "Pedidos Distri Romel SAS", se puede apreciar cómo van llegando los pedidos realizados por los Tenderos, junto con sus datos personales, para poder contactarlos en caso de que sea requerido, el total a pagar y la lista de productos que cada uno de estos ordeno, junto con sus cantidades y descripciones, como se había solicitado por el mismo Proveedor. En esta vista, el Chatbot también informará al Grupo cuando hay escasez de un producto, para que este pueda ser abastecido nuevamente en su inventario.

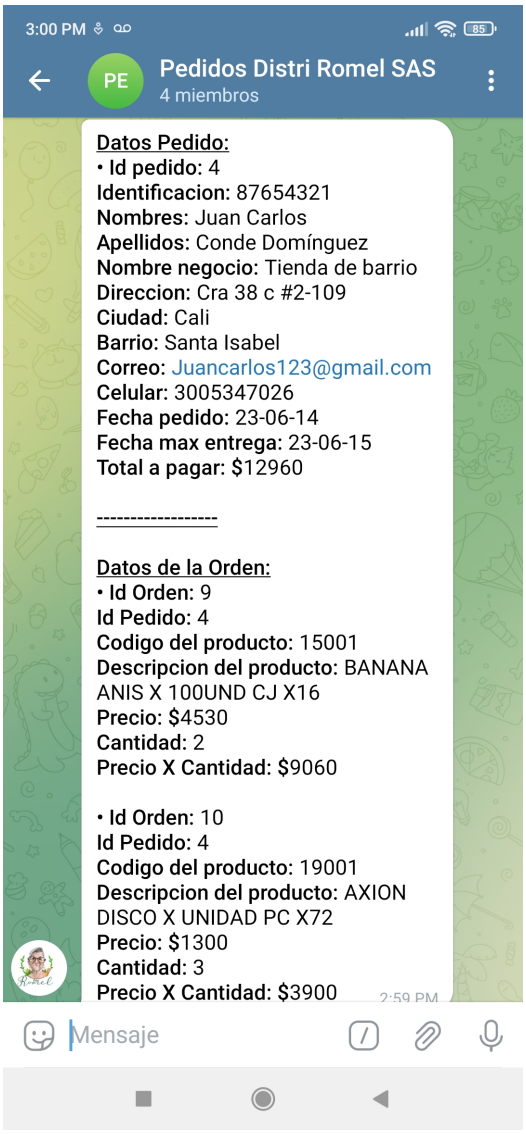


Figura 10.37: Vista pedido recibido Proveedor Parte II.

10.4. Formulario Retroalimentación Chatbot Usuario Tendero

- En la Figura 10.38, se puede visualizar el formulario de la encuesta realizada a la población de estudio que interactuó con el rol de usuario Tendero. En él se solicita la edad del encuestado y se le pregunta qué tipo de plan de datos tiene.

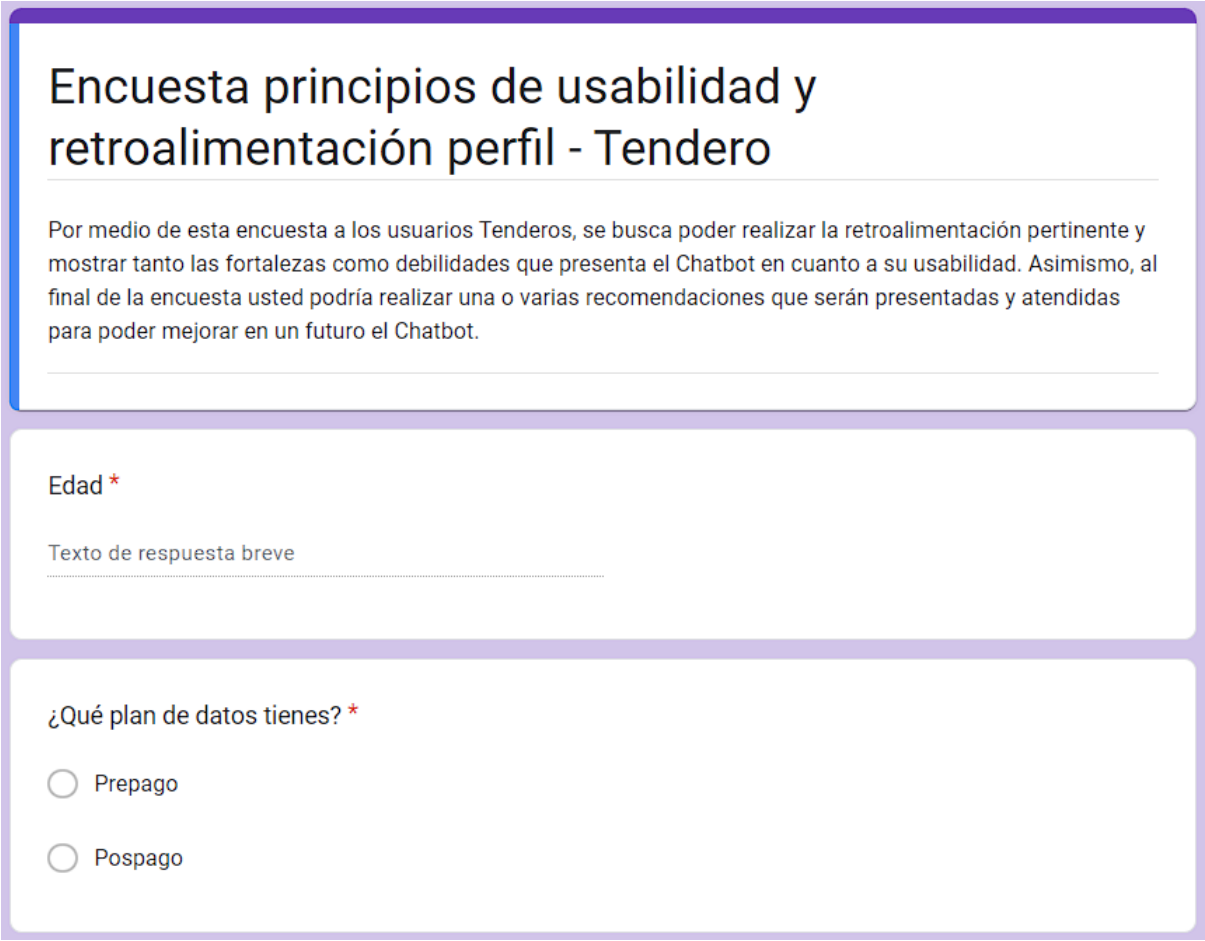


Figura 10.38: Encuesta Usuario Tendero Parte I.

- En la Figura 10.39, se puede visualizar el formulario de la encuesta realizada a la población de estudio que interactuó con el rol de usuario Tendero. En ella, se realizan tres preguntas alusivas a los principios número 1, 2 y 3 de usabilidad, de Jakob Nielsen.

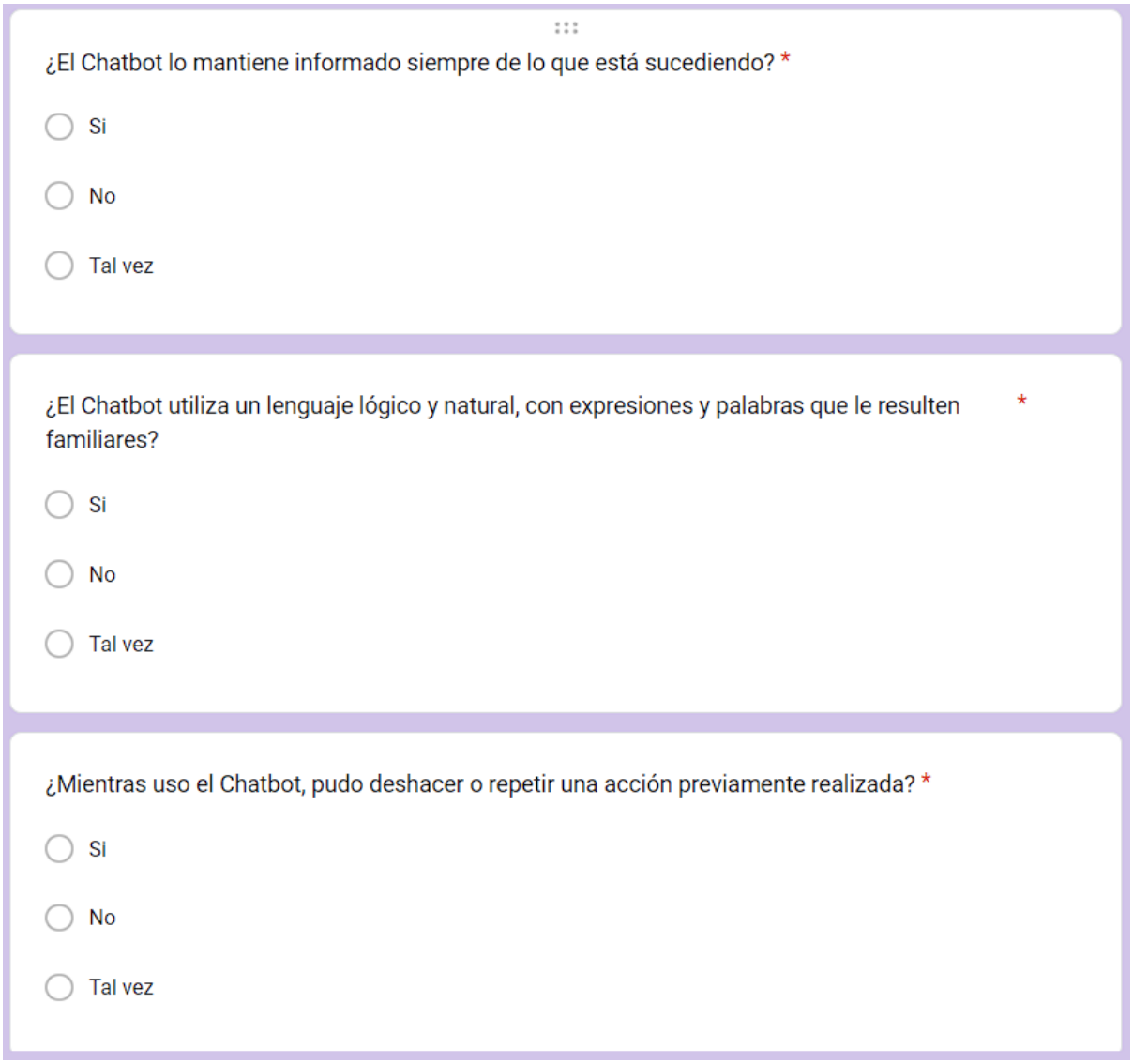


Figura 10.39: Encuesta Usuario Tendero Parte II.

- En la Figura 10.40, se puede visualizar el formulario de la encuesta realizada a la población de estudio que interactuó con el rol de usuario Tendero. En ella, se realizan tres preguntas alusivas a los principios número 4, 5 y 6 de usabilidad, de Jakob Nielsen.

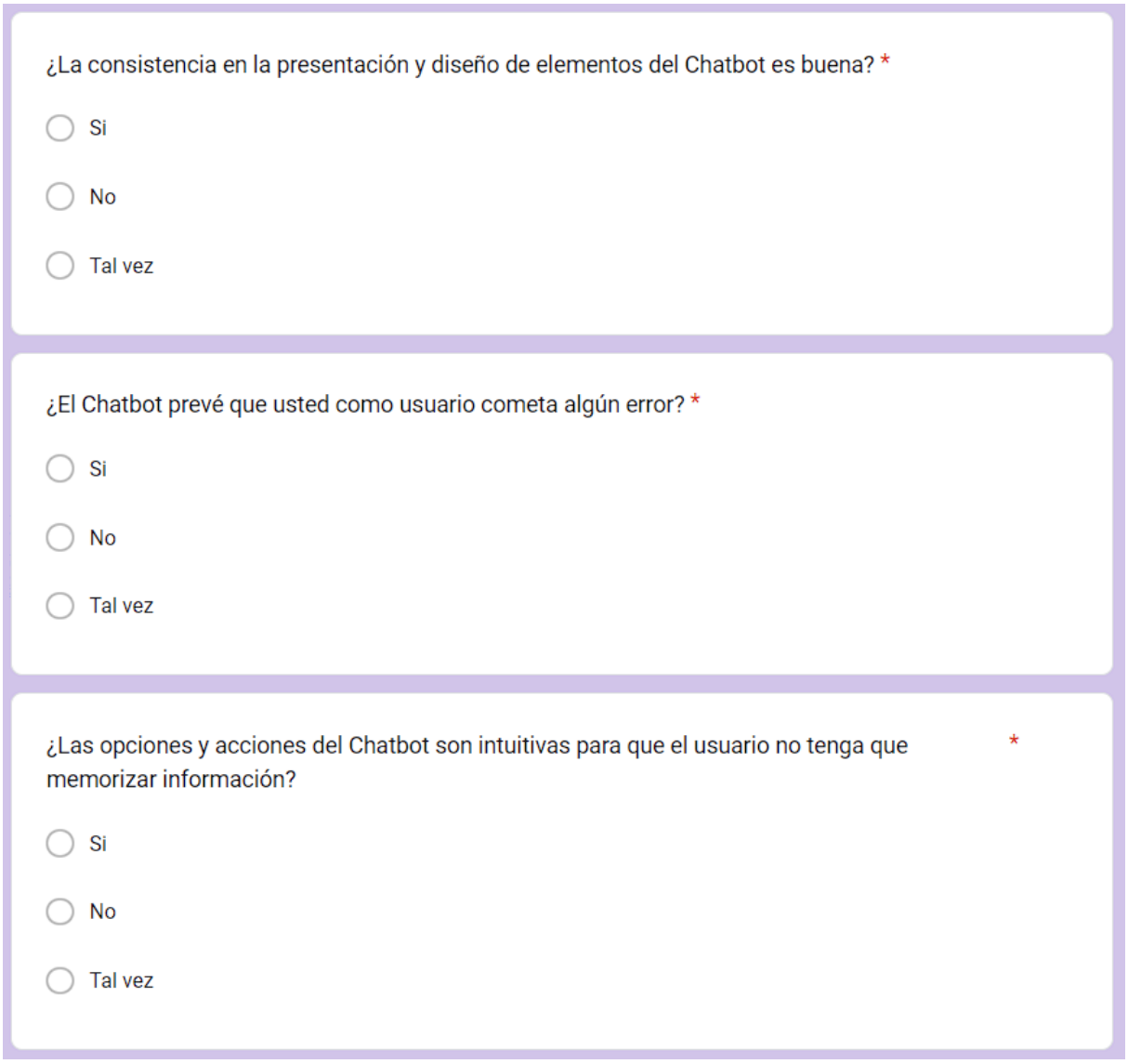


Figura 10.40: Encuesta Usuario Tendero Parte III.

- En la Figura 10.41, se puede visualizar el formulario de la encuesta realizada a la población de estudio que interactuó con el rol de usuario Tendero. En ella, se realizan tres preguntas alusivas a los principios número 7, 8 y 9 de usabilidad, de Jakob Nielsen.

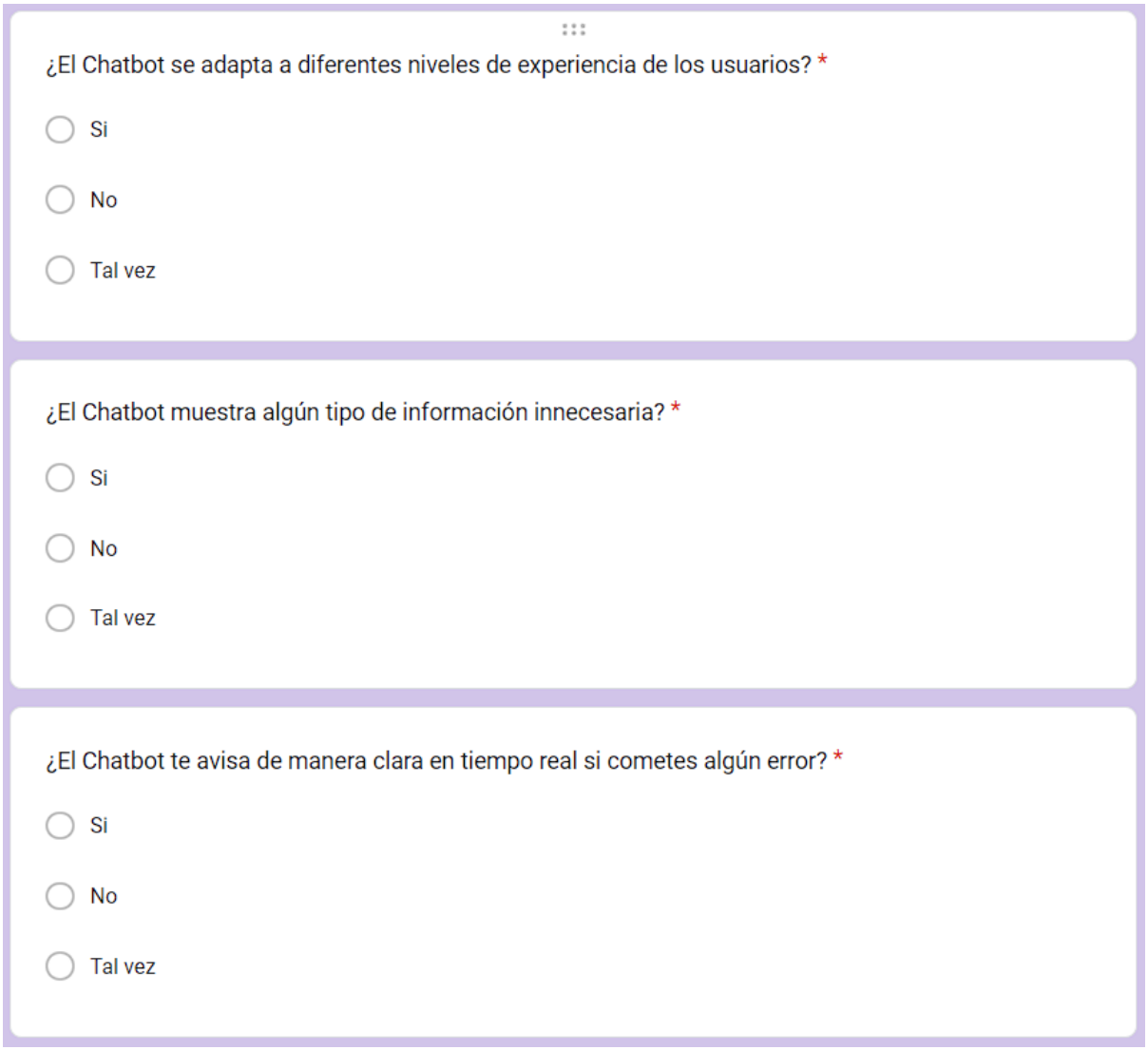


Figura 10.41: Encuesta Usuario Tendero Parte IV.

- En la Figura 10.42, se puede visualizar el formulario de la encuesta realizada a la población de estudio que interactuó con el rol de usuario Tendero. En ella, se formula una pregunta alusiva al principio número 10 de usabilidad de Jakob Nielsen. Además, se cuenta con una pregunta adicional abierta donde el usuario puede escribir alguna sugerencia o recomendación para mejorar el Chatbot.

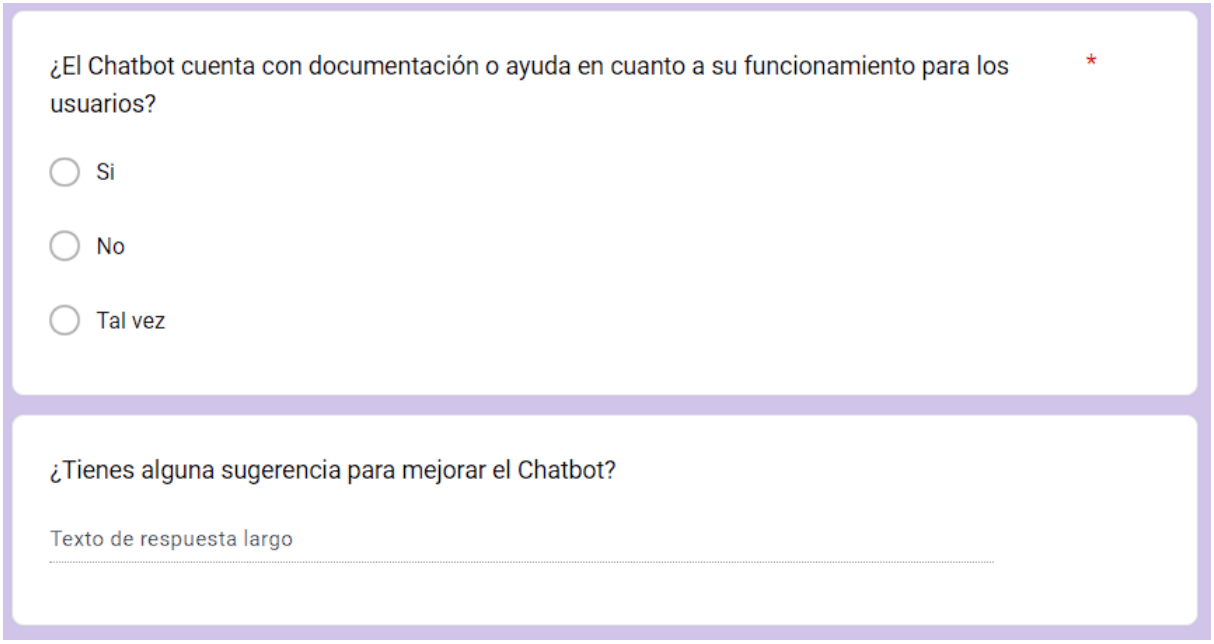


Figura 10.42: Encuesta Usuario Tendero Parte V.

10.5. Formulario Retroalimentación Chatbot Usuario Proveedor

En la Figura 10.43, se puede visualizar el formulario de la encuesta realizada a la población de estudio que interactuó con el rol de usuario Proveedor. En él, se hacen dos preguntas alusivas a los principios número 1 y 2 de usabilidad, de Jakob Nielsen.

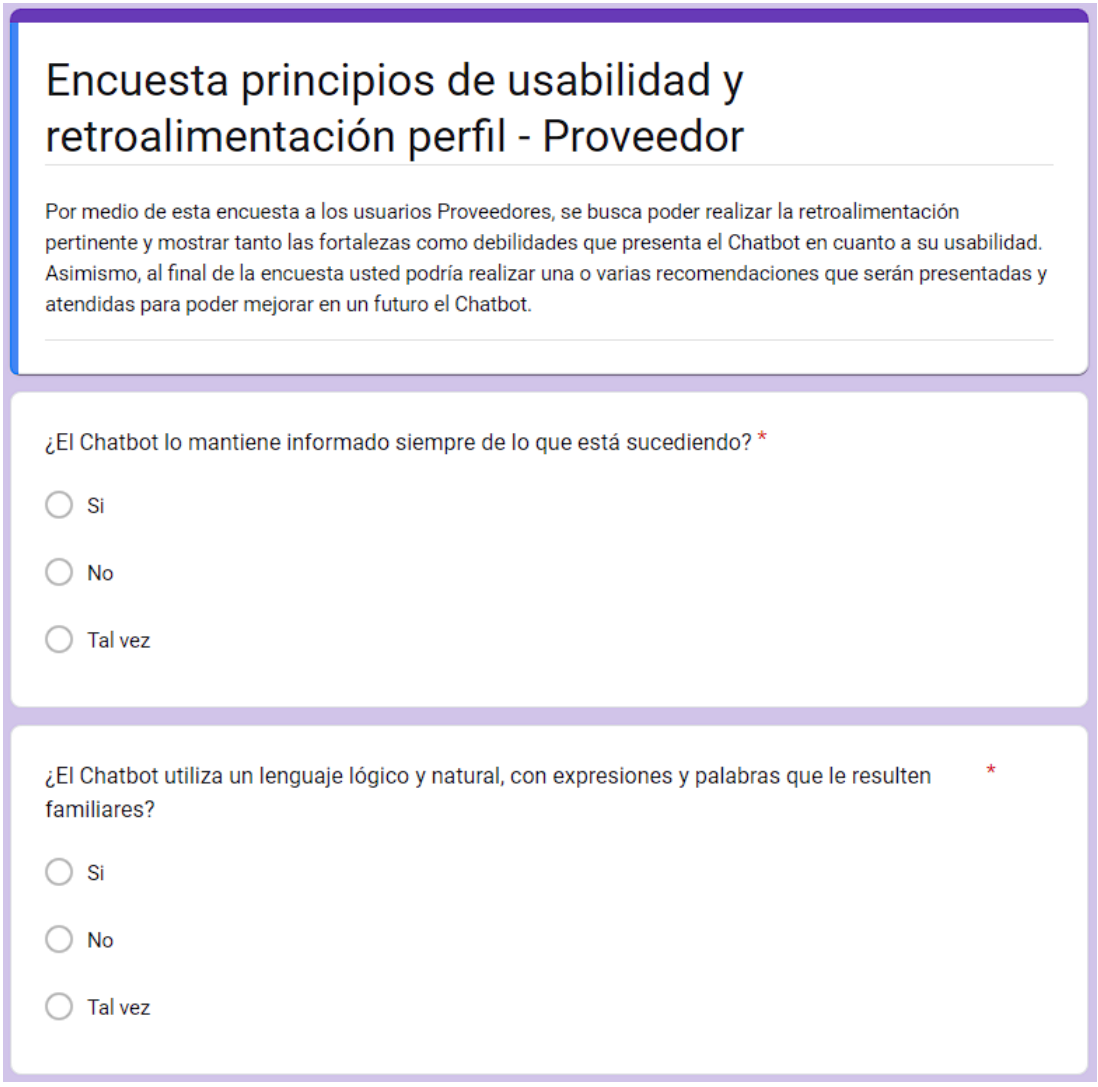


Figura 10.43: Encuesta Usuario Proveedor Parte I.

En la Figura 10.44, se puede visualizar el formulario de la encuesta realizada a la población de estudio que interactuó con el rol de usuario Proveedor. En él, se hacen tres preguntas alusivas a los principios número 3, 4 y 5 de usabilidad, de Jakob Nielsen.

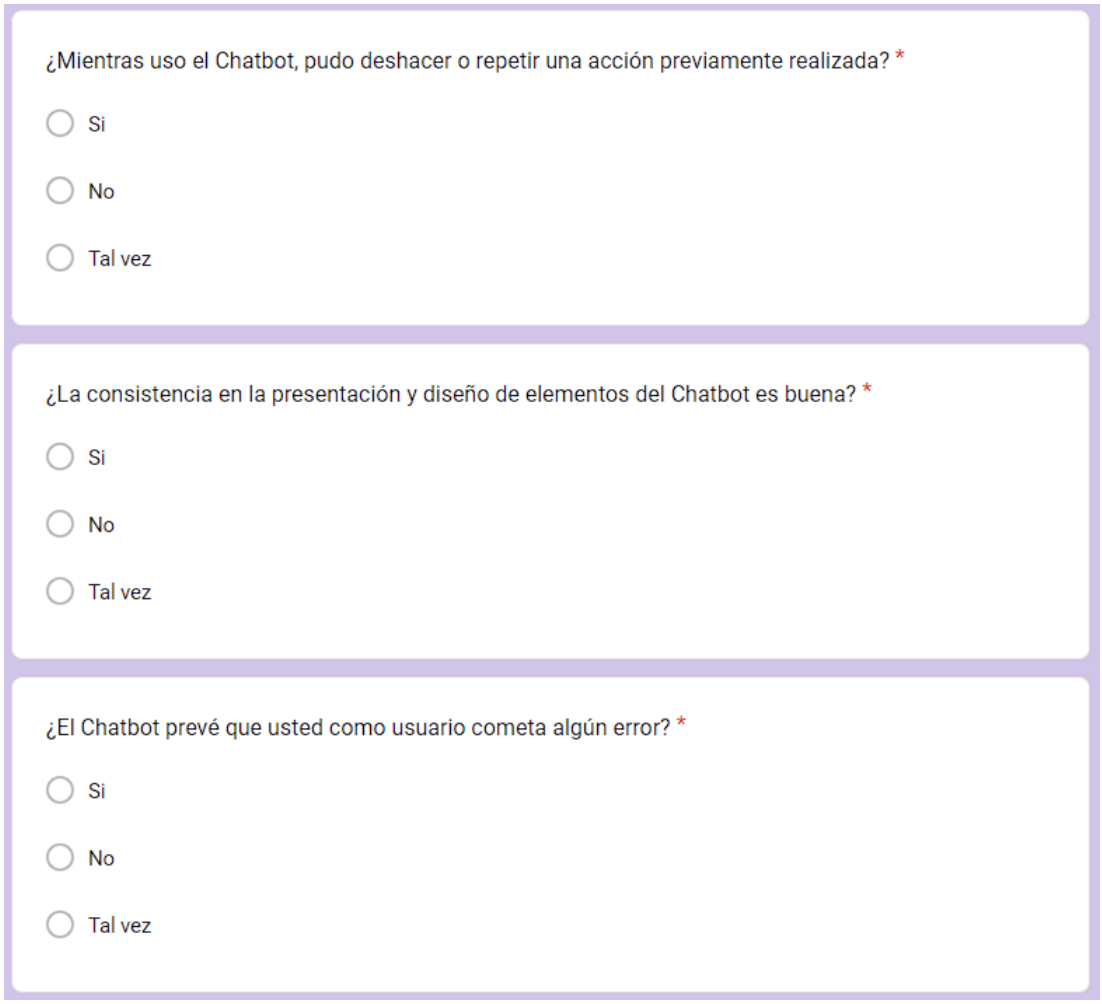


Figura 10.44: Encuesta Usuario Proveedor Parte II.

En la Figura 10.45, se puede visualizar el formulario de la encuesta realizada a la población de estudio que interactuó con el rol de usuario Proveedor. En él, se hacen tres preguntas alusivas a los principios número 6, 7 y 8 de usabilidad, de Jakob Nielsen.

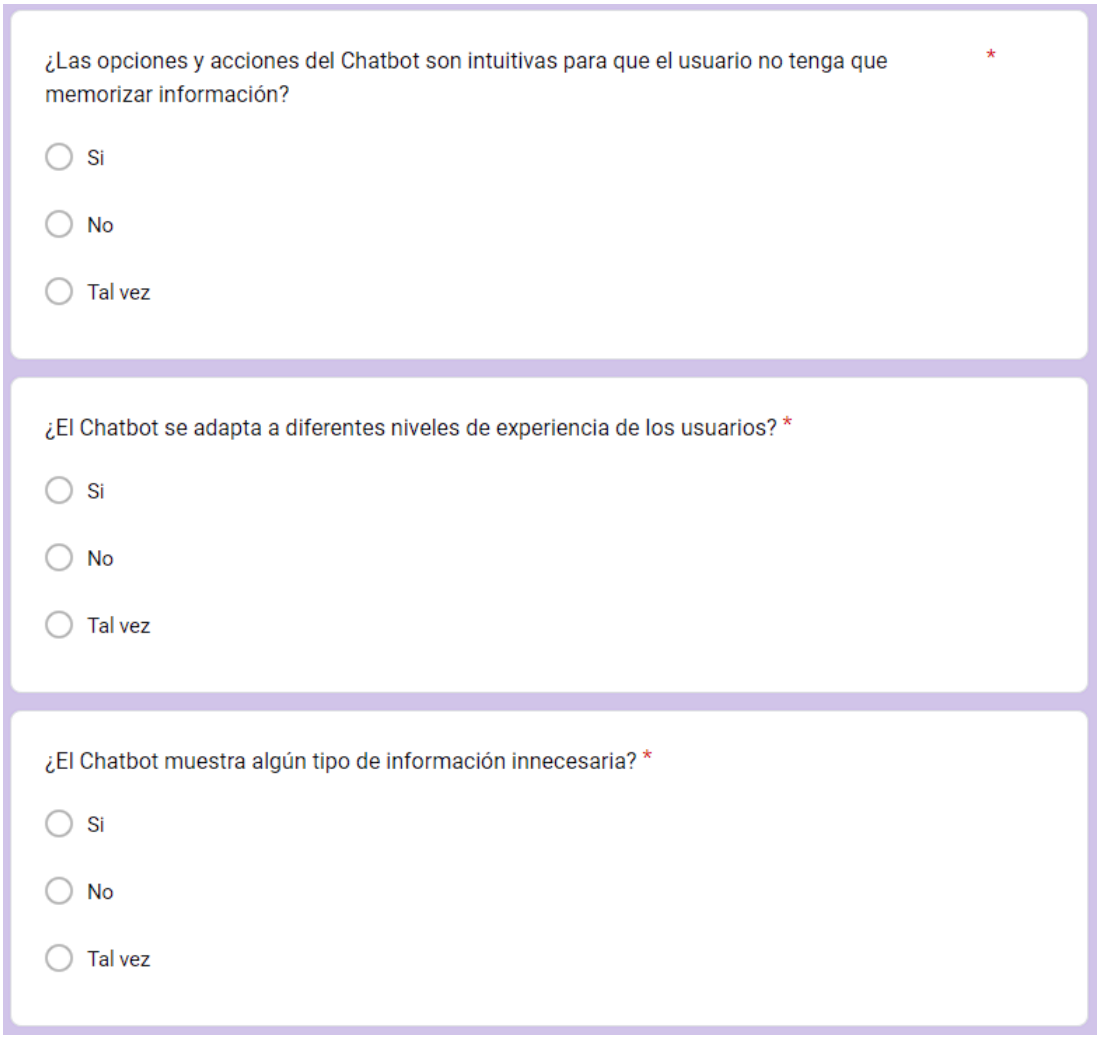


Figura 10.45: Encuesta Usuario Proveedor Parte III.

En la Figura 10.46, se puede visualizar el formulario de la encuesta realizada a la población de estudio que interactuó con el rol de usuario Proveedor. En él, se hacen dos preguntas alusivas a los principios número 9 y 10 de usabilidad, de Jakob Nielsen. Además, se cuenta con una pregunta adicional abierta donde el usuario puede escribir alguna sugerencia o recomendación para mejorar el Chatbot.

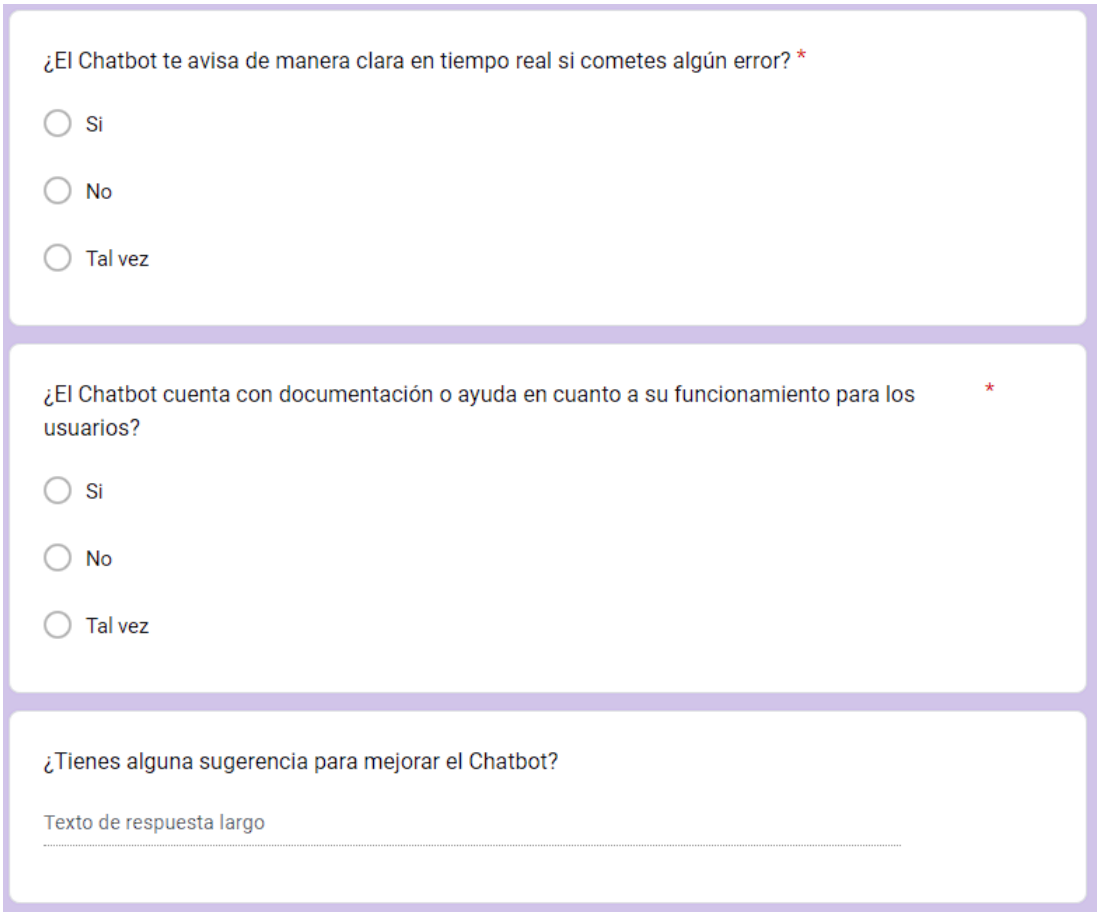


Figura 10.46: Encuesta Usuario Proveedor Parte IV.

10.6. Casos de Prueba

A continuación, se visualizarán todos los casos de prueba que se han considerado para realizar las pruebas funcionales pertinentes al desarrollo del Chatbot.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Iniciar una conversación con el chatbot en Telegram.	
Identificador	CP_FUN_001	
Nombre del Caso	Prueba de bienvenida	
Precondiciones	• Encontrarse dentro de la aplicación de Telegram	
Paso	**Resultado esperado**	**Resultado real**
1) Ir a la opcion "buscar" y buscar el chatbot nombrado "Distri Romel"	El chatbot debe enviar un mensaje de bienvenida y proporcionar opciones o instrucciones claras para el usuario.	**OK**
2) Ingresar en la conversacion y presionar la opcion "iniciar"		

Figura 10.47: Caso de prueba CP_FUN_001.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Enviar comandos básicos como "/inicio" o "/ayuda" al chatbot.	
Identificador	CP_FUN_002	
Nombre del Caso	Prueba de comandos básicos	
Precondiciones	• Encontrarse dentro de la conversacion con el Chatbot	
Paso	**Resultado esperado**	**Resultado real**
1) Escribir el comando "inicio" o "ayuda" y enviar el mensaje	El chatbot debe responder con información de ayuda o volver a mostrar el mensaje de bienvenida.	**OK**

Figura 10.48: Caso de prueba CP_FUN_002.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Enviar un comando que el chatbot no reconozca.	
Identificador	CP_FUN_003	
Nombre del Caso	Prueba de comandos desconocidos	
Precondiciones	• Encontrarse dentro de la conversacion con el Chatbot	
Paso	**Resultado esperado**	**Resultado real**
1) Escribir un comando distinto y enviar el mensaje	El chatbot debe responder con un mensaje indicando que no comprende el comando y, si es relevante, ofrecer una lista de comandos válidos.	**OK**

Figura 10.49: Caso de prueba CP_FUN_003.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Hacer clic en un enlace o botón interactivo proporcionado por el chatbot.	
Identificador	CP_FUN_004	
Nombre del Caso	Prueba de enlaces y botones interactivos	
Precondiciones	• Encontrarse dentro de la conversacion con el Chatbot	
Paso	**Resultado esperado**	**Resultado real**
1) Presionar un boton o enlace durante la conversacion con el Chatbot	El chatbot debe dirigir al usuario a la página web correspondiente o realizar la acción asociada al botón.	**OK**

Figura 10.50: Caso de prueba CP_FUN_004.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Responder a una pregunta o solicitud del usuario.	
Identificador	CP_FUN_005	
Nombre del Caso	Prueba de interacción de usuario	
Precondiciones	• Encontrarse dentro de la conversacion con el Chatbot	
Paso	**Resultado esperado**	**Resultado real**
1) Presionar un boton o enviar un mensaje al Chatbot	El chatbot debe procesar la respuesta del usuario y proporcionar una respuesta relevante y coherente.	**OK**

Figura 10.51: Caso de prueba CP_FUN_005.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Validar boton ayuda	
Identificador	CP_FUN_006	
Nombre del Caso	Prueba de interaccion con el boton Ayuda	
Precondiciones	• Encontrarse dentro de la conversacion con el Chatbot	
Paso	**Resultado esperado**	**Resultado real**
1) Presionar el boton de Ayuda en la conversacion con el Chatbot	El chatbot debe dirigir a la seccion correspondiente al usuario y enviarle el link del video explicativo.	**OK**

Figura 10.52: Caso de prueba CP_FUN_006.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Validar boton ayuda	
Identificador	CP_FUN_007	
Nombre del Caso	Prueba de interaccion con el boton Contacto	
Precondiciones	• Encontrarse dentro de la conversacion con el Chatbot	
Paso	**Resultado esperado**	**Resultado real**
1) Presionar el boton de Contacto en la conversacion con el Chatbot	El chatbot debe dirigir a la seccion correspondiente al usuario y enviarle la informacion relevante del proveedor.	**OK**

Figura 10.53: Caso de prueba CP_FUN_007.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Validar Registro Usuario Tendero	
Identificador	CP_REG_001	
Nombre del Caso	Prueba de registro usuario tendero	
Precondiciones	• Encontrarse dentro de la conversacion con el Chatbot • No haberse registrado previamente.	
Paso	**Resultado esperado**	**Resultado real**
1) Presionar el boton de "Realizar Pedido"	El chatbot debe procesar los datos ingresados por el usuario y enviar una respuesta que indique que se ha registrado exitosamente	**OK**
2) Ingresar los datos solicitados por el Chatbot		

Figura 10.54: Caso de prueba CP_REG_001.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Validar Proceso Realizar Pedido	
Identificador	CP_PED_001	
Nombre del Caso	Prueba de validacion usuario tendero	
Precondiciones	• Encontrarse dentro de la conversacion con el Chatbot • Haberse registrado previamente.	
Paso	**Resultado esperado**	**Resultado real**
1) Presionar el boton de "Realizar Pedido"	El chatbot debe validar que el usuario se encuentra registrado en la BD y habilitar o responder con un boton que va a permitir abrir la Web App para realizar el pedido	**OK**
2) Ingresar la identificacion solicitada por el Chatbot		

Figura 10.55: Caso de prueba CP_PED_001.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Validar Proceso Realizar Pedido	
Identificador	CP_PED_002	
Nombre del Caso	Prueba funcionamiento boton realizar pedido	
Precondiciones	• Encontrarse dentro de la conversacion con el Chatbot • Haberse registrado previamente • Haber pasado la validacion de identificacion	
Paso	**Resultado esperado**	**Resultado real**
1) Presionar el nuevo boton que se habilita para realizar un pedido	El chatbot debe abrir la Web App que cuenta con un catalogo de productos y un carrito para que el usuario pueda realizar el pedido	**OK**

Figura 10.56: Caso de prueba CP_PED_002.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Validar Proceso Realizar Pedido	
Identificador	CP_PED_003	
Nombre del Caso	Prueba funcionamiento añadir productos al carrito	
Precondiciones	• Encontrarse dentro de la conversacion con el Chatbot • Haberse registrado previamente • Haber pasado la validacion de identificacion • Haber abierto la Web App	
Paso	**Resultado esperado**	**Resultado real**
1) Seleccionar una categoria en el catalogo de productos	El chatbot dentro de la Web App debe permitir añadir un producto al carrito de compras	**OK**
2) Presionar el boton sobre un producto para añadirlo al carrito		

Figura 10.57: Caso de prueba CP_PED_003.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Validar Proceso Realizar Pedido	
Identificador	CP_PED_004	
Nombre del Caso	Prueba funcionamiento realizar pedido	
Precondiciones	• Encontrarse dentro de la conversacion con el Chatbot • Haberse registrado previamente • Haber pasado la validacion de identificacion • Haber abierto la Web App • Haber añadido algun producto al carrito de compras	
Paso	**Resultado esperado**	**Resultado real**
1) Abrir el Carrito de compras	El chatbot dentro de la Web App debe permitir realizar el pedido exitosamente y enviar los datos del pedido al proveedor	**OK**
2) Presionar el boton "Realizar pedido" en la parte inferior del carrito de compras		

Figura 10.58: Caso de prueba CP_PED_004.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Validar Proceso Realizar Pedido	
Identificador	CP_PED_005	
Nombre del Caso	Prueba envio pedido grupo Telegram	
Precondiciones	• Que un usuario tendero haya realizado un pedido • Como proveedor estar agregado al grupo de Telegram en el que esta el Chatbot • encontrarse en el inicio de la aplicacion de Telegram	
Paso	**Resultado esperado**	**Resultado real**
1) Abrir la conversacion o grupo de Telegram	El chatbot debe enviar al grupo de telegram el pedido realizado por el tendero junto con sus datos de contacto relevantes	**OK**

Figura 10.59: Caso de prueba CP_PED_005.

Formato de Caso de Prueba		
Objetivo del Caso de Prueba	Validar Proceso Realizar Pedido	
Identificador	CP_PED_006	
Nombre del Caso	Prueba envio notificacion productos grupo Telegram	
Precondiciones	• Que un usuario tendero haya realizado un pedido • Como proveedor estar agregado al grupo de Telegram en el que esta el Chatbot • encontrarse en el inicio de la aplicacion de Telegram	
Paso	**Resultado esperado**	**Resultado real**
1) Abrir la conversacion o grupo de Telegram	El chatbot debe enviar al grupo de telegram una notificacion avisando cuando hay escasez de un producto en el inventario	**OK**

Figura 10.60: Caso de prueba CP_PED_006.