

# Informe de Diseño y Desarrollo: Bot Conversacional de WhatsApp para ProAlto

## 1.0 Introducción y Contexto Estratégico

### 1.1 Visión General y Objetivos del Proyecto

Este informe presenta una estrategia integral para el diseño, desarrollo e implementación de un bot conversacional de WhatsApp para ProAlto. En el competitivo sector de créditos de libranza en Colombia, la adopción de esta tecnología no es solo una mejora operativa, sino una necesidad estratégica fundamental. La implementación de un asistente virtual inteligente permitirá a ProAlto escalar su operación, optimizar la eficiencia de su centro de atención y satisfacer las crecientes expectativas de inmediatez y disponibilidad del cliente digital moderno. Los objetivos clave que guían este proyecto son los siguientes:

- **Automatizar la primera línea de interacción** con el cliente, resolviendo consultas frecuentes y transacciones rutinarias sin necesidad de intervención humana, liberando al equipo para gestionar casos de mayor complejidad.
- **Mejorar la disponibilidad del servicio** a un modelo 24/7, permitiendo a empleados y pensionados realizar consultas y gestionar sus créditos en cualquier momento, incluso fuera del horario de oficina tradicional.
- **Reducir los costos operativos** asociados al centro de atención al cliente, disminuyendo la carga de trabajo en tareas repetitivas y optimizando la asignación de recursos humanos.
- **Acelerar los tiempos de respuesta** a consultas críticas, pasando de horas a segundos, lo que se traduce en un aumento significativo de la satisfacción y lealtad del cliente.
- **Centralizar y estandarizar la comunicación**, garantizando que cada respuesta sea consistente, precisa y esté perfectamente alineada con la voz y los estándares de la marca ProAlto. Para alcanzar estos objetivos, es indispensable partir de un entendimiento profundo del cliente. El siguiente análisis del perfil del usuario y sus necesidades prioritarias es la piedra angular sobre la cual se construirá toda la experiencia conversacional.

### 1.2 Perfil del Cliente y Casos de Uso Prioritarios

El diseño de una experiencia conversacional efectiva comienza y termina con el cliente. El usuario de ProAlto es principalmente un empleado o pensionado que ha optado por un crédito de libranza, un modelo de préstamo cuyo pago se descuenta directamente de su nómina o mesada pensional. Este perfil valora la simplicidad, la claridad y, sobre todo, la inmediatez en la gestión de sus finanzas. Esperan respuestas rápidas y seguras que les permitan tener control sobre su estado crediticio sin fricciones ni demoras. El cliente de ProAlto busca resolver dudas puntuales de manera autónoma, prefiriendo un canal directo y accesible como WhatsApp antes que una llamada telefónica. Sus interacciones más comunes han sido identificadas y priorizadas para ser el núcleo funcional del bot. A continuación, se presentan los casos de uso prioritarios que el bot debe resolver de manera autónoma:

Caso de Uso	Objetivo del Bot
Consulta Frecuente del Cliente	Proveer de forma segura el saldo actual del crédito, el número de cuotas pagadas/pendientes y la fecha del próximo descuento de nómina.
Consulta de Saldos	
Estado de la Solicitud	

| Informar en tiempo real en qué etapa se encuentra una nueva solicitud de crédito (recepción, estudio, aprobación, desembolso). || **Generación de Paz y Salvo** | Automatizar la creación y envío inmediato del certificado de paz y salvo una vez la obligación ha sido completamente cancelada. || **Solicitud de Renovación** | Facilitar el inicio del proceso para una ampliación o novación del crédito actual, guiando al cliente en los primeros pasos. || **Nueva Solicitud de Crédito** | Guiar a un cliente potencial a través de los requisitos iniciales, la documentación necesaria y los pasos para iniciar una solicitud de crédito. | La comprensión detallada de estas necesidades nos permite ahora diseñar una arquitectura funcional y flujos conversacionales que no solo respondan, sino que anticipen las expectativas del cliente.

## 2.0 Arquitectura Funcional y Flujos Conversacionales

### 2.1 Diseño de la Conversación

El éxito de este proyecto no reside en la tecnología por sí misma, sino en la calidad de la conversación. Un flujo de diálogo lógico, empático y eficiente es clave para que el bot sea percibido como una herramienta genuinamente útil y no como un obstáculo burocrático. El diseño debe guiar al usuario de manera intuitiva hacia la resolución de su consulta, minimizando la cantidad de pasos y la fricción en cada interacción. El flujo de bienvenida establecerá el tono de la experiencia. Al recibir el primer mensaje, el bot saludará cordialmente, se presentará como el "Asistente Virtual de ProAlto" y ofrecerá un menú principal con botones de respuesta rápida para acceder directamente a las cinco consultas prioritarias, evitando que el usuario tenga que adivinar comandos. A continuación, se describen los flujos conversacionales para cada caso de uso:

1. **Consulta de Saldos:**
2. El usuario **selecciona la opción** "Consultar mi saldo".
3. El bot **solicita un método de validación de identidad** para proteger la información sensible (ej. últimos dígitos de la cédula o un código OTP enviado al móvil registrado).
4. Tras la validación exitosa, el bot **ejecuta una consulta segura** a la base de datos de ProAlto.
5. El bot **presenta la información** de manera clara y concisa: saldo pendiente, número de cuotas pagadas/totales y fecha del próximo descuento.
6. Finalmente, **pregunta de forma proactiva** : "¿Hay algo más en lo que pueda ayudarte?".
7. **Estado de la Solicitud:**
8. El usuario **elige** "Estado de mi solicitud".
9. El bot **solicita el número de cédula** para identificar la solicitud.
10. El sistema **consulta el estado actual** del trámite.
11. El bot **informa la etapa** en la que se encuentra la solicitud (ej. "Tu solicitud de crédito está en etapa de estudio y recibirás una respuesta en las próximas 48 horas.").
12. **Ofrece la opción de ser notificado** automáticamente cuando haya un cambio de estado.
13. **Generación de Paz y Salvo:**
14. El usuario **selecciona** "Generar Paz y Salvo".
15. El bot **valida la identidad** del usuario.

16. **Verifica en el sistema** que todas las obligaciones del cliente con ProAlto estén canceladas.
17. Si todo está en orden, **genera automáticamente el certificado** en formato PDF y lo envía directamente al chat del usuario.
18. Si aún hay saldos pendientes, **informa al usuario** de manera clara y ofrece la opción de consultar el detalle.
19. **Solicitud de Renovación:**
20. El usuario **elige** "Solicitar una renovación".
21. El bot **valida la identidad y consulta la elegibilidad** del cliente para una renovación.
22. Si es elegible, **presenta una oferta preaprobada** o informa sobre los documentos necesarios para iniciar el estudio (ej. últimos desprendibles de pago).
23. **Permite al usuario cargar los documentos** requeridos directamente en el chat.
24. **Confirma la recepción y deriva la solicitud** a un asesor comercial para su finalización.
25. **Nueva Solicitud de Crédito:**
26. El usuario **selecciona** "Solicitar un nuevo crédito".
27. El bot **presenta los requisitos básicos** y la documentación necesaria (fotocopia de cédula, certificación de ingresos, etc.).
28. **Permite al prospecto cargar los documentos** directamente como imágenes o archivos PDF.
29. **Recopila datos de contacto** básicos.
30. **Asigna la solicitud a un asesor comercial** y le informa al prospecto que será contactado en breve. Dada la naturaleza del negocio, el bot debe estar preparado para **manejar documentos**, permitiendo la recepción segura de archivos como fotocopias de cédula o certificados de ingresos en formatos estándar (PDF, JPG, PNG). Aunque el objetivo es maximizar la automatización, es crucial definir un plan claro para escalar las conversaciones a un agente humano cuando la situación lo requiera.

## 2.2 Escalación a Agente Humano y Seguridad

Si bien la automatización es el pilar de este proyecto, una transición fluida y contextualizada hacia un agente humano es crucial para resolver casos complejos, gestionar situaciones emocionales y evitar la frustración del cliente. El bot debe ser un facilitador, no una barrera.

Por ello, el protocolo de escalación es una parte integral del diseño. Los **disparadores clave para la escalación humana** serán:

- **Fallo de Comprensión:** Cuando el bot no logra entender la consulta del usuario después de dos intentos consecutivos.
- **Solicitud Explícita:** Cuando el usuario escribe frases como "hablar con un asesor", "ayuda humana" o "necesito un agente".
- **Consultas Complejas:** Para situaciones que por su naturaleza requieren empatía y análisis humano, como quejas, reclamos o casos de fraude. En cuanto a la **seguridad**, es un pilar no negociable. Antes de entregar cualquier tipo de información financiera sensible, el bot implementará un **protocolo de validación de identidad de múltiples factores**. Este proceso confirmará que se está interactuando con el titular del crédito, cumpliendo rigurosamente con las normativas de protección de datos personales y garantizando la confidencialidad de la

información. Con la arquitectura funcional definida, el siguiente paso es analizar las dos opciones técnicas principales para llevar este diseño a la realidad.

### 3.0 Análisis Comparativo de Opciones de Desarrollo

La implementación de este bot conversacional puede abordarse a través de dos vías estratégicas principales: el uso de una plataforma "No-Code" para un despliegue rápido y una gestión visual, o un "Desarrollo a Medida" que ofrece máxima personalización y control sobre la infraestructura. La elección dependerá del balance deseado entre velocidad de implementación, flexibilidad y control a largo plazo. A continuación, se presenta una tabla comparativa para evaluar ambas opciones:

Criterio	Opción 1: Plataforma No-Code	Opción 2: Desarrollo a Medida
<b>Tiempo de implementación</b>	<b>Rápido</b> (Semanas).	<b>Moderado a Lento</b> (Meses).
<b>Configuración visual e interfaces preconstruidas</b>	<b>Alto</b>	<b>Bajo</b>
<b>Requiere ciclo completo de desarrollo de software</b>	<b>Bajo</b>	<b>Alto</b>
<b>Costo inicial y recurrente</b>	<b>Bajo</b> costo inicial. Modelo de suscripción mensual (SaaS) predecible.	<b>Alto</b> costo inicial (desarrollo). Costo recurrente menor (solo infraestructura).
<b>Flexibilidad y personalización</b>	<b>Media</b> . Limitada a las funcionalidades ofrecidas por la plataforma.	<b>Alta</b> . Control total sobre la lógica, integraciones y experiencia de usuario.
<b>Facilidad de mantenimiento</b>	<b>Alta</b> . La plataforma gestiona la infraestructura y las actualizaciones.	<b>Media a Baja</b> . Requiere un equipo técnico para mantenimiento y actualizaciones.
<b>Escalabilidad</b>	<b>Buena</b> . Depende del plan contratado y las capacidades de la plataforma.	<b>Excelente</b> . Diseñada para escalar según las necesidades específicas de ProAlto.
<b>Control sobre los datos</b>	<b>Limitado</b> . Los datos de la conversación residen en la infraestructura del proveedor.	<b>Total</b> . Los datos se alojan en la infraestructura propia o controlada por ProAlto.

Para una validación rápida del concepto y un menor costo inicial, la opción No-Code es preferible. Para un control total y una máxima escalabilidad a futuro, el Desarrollo a Medida es la inversión estratégica correcta. Las siguientes secciones detallarán una guía de implementación para cada una de estas opciones, permitiendo a ProAlto tomar una decisión informada basada en sus prioridades estratégicas.

### 4.0 Opción 1: Desarrollo No-Code con Plataforma Visual

#### 4.1 Selección de Herramienta: ManyChat

La primera opción se centra en utilizar una plataforma de desarrollo No-Code, un enfoque que prioriza la velocidad de implementación y reduce significativamente la barrera de entrada técnica. Esta vía permite que equipos no especializados en programación puedan diseñar, construir y mantener flujos conversacionales complejos a través de una interfaz gráfica e intuitiva. Para este enfoque, se recomienda **ManyChat** como la herramienta principal. Esta elección se justifica por su robusta interfaz visual de arrastrar y soltar, su popularidad como punto de partida para empresas que inician en la automatización de mensajería y su capacidad para integrarse con sistemas externos mediante solicitudes HTTP, lo cual es esencial para conectar con la base de datos de ProAlto. ManyChat es ideal para un despliegue rápido y una validación ágil del modelo. A continuación, se presenta una guía detallada para que un desarrollador implemente el bot de ProAlto utilizando esta plataforma.

## 4.2 Guía Detallada de Implementación en ManyChat

- **Creación de la Cuenta y Configuración Inicial:**
  - Registrarse en [manychat.com](https://manychat.com) utilizando una cuenta de empresa.
  - Dentro del panel de control, seleccionar **WhatsApp** como el canal principal a configurar.
- **Conexión con la API de WhatsApp Business:**
  - Navegar a Settings → WhatsApp y hacer clic en "Connect".
  - Seguir el asistente de configuración de Meta, que requerirá vincular una cuenta de **Facebook Business Manager** previamente verificada.
  - Proporcionar y verificar un número de teléfono dedicado exclusivamente para el bot. Es crucial utilizar la **API oficial de WhatsApp Business** a través de este proceso para garantizar el cumplimiento de las políticas y evitar bloqueos.
- **Diseño del Flujo de Bienvenida:**
  - Ir a Automation → Flows y crear un nuevo flujo.
  - Establecer el disparador (trigger) para que se active cuando un usuario envía su primer mensaje.
  - Diseñar el mensaje de saludo inicial, presentándose como el Asistente Virtual de ProAlto.
  - Añadir un bloque de **botones de respuesta rápida** con las cinco funciones clave: "Consultar Saldo", "Estado de Solicitud", "Generar Paz y Salvo", "Solicitar Renovación" y "Nueva Solicitud de Crédito".
- **Construcción de Flujos para Consultas Frecuentes:**
  - Para cada botón del menú principal, crear una rama lógica (un nuevo flujo o una secuencia de pasos) que gestione la conversación.
  - Utilizar bloques de "User Input" para solicitar información al usuario (ej. número de cédula) y guardarla en variables.
  - Utilizar bloques de "Condition" para validar la información recibida.
- **Integración con Base de Datos PostgreSQL:**
  - En los flujos que requieran datos del cliente (saldos, estado de solicitud), utilizar el bloque **External Request (Solicitud HTTP)**.
  - Configurar este bloque para que realice una llamada POST a un punto final (API) seguro que el equipo de desarrollo de ProAlto debe construir.
  - Esta API interna será responsable de:
    - Recibir los datos de validación del bot (ej. cédula del cliente).
    - Realizar la consulta segura a la base de datos PostgreSQL.
    - Devolver la información requerida en formato JSON.
  - Mapear la respuesta JSON del API a variables de ManyChat para presentar la información al usuario.
- **Configuración de la Derivación a Humanos:**
  - Ir a Settings → Live Chat para configurar los horarios de atención humana.
  - En los flujos conversacionales, añadir una opción de "Hablar con un asesor" que active la acción "Notify Admins" y abra una sesión de chat en vivo.
  - Configurar reglas para que, si el bot no entiende la consulta del usuario, se active automáticamente la derivación a un agente.
- **Pruebas y Publicación:**
  - Utilizar la función de "Preview" para probar exhaustivamente todos los flujos desde un número de WhatsApp de prueba.

- Verificar que las integraciones con la API externa funcionen correctamente y que los datos se muestren de forma precisa.
- Una vez validados todos los flujos, activar el bot desde el panel de control para que comience a interactuar con los clientes. Este enfoque No-Code ofrece una ruta rápida y eficiente hacia la automatización. A continuación, exploraremos la segunda opción, un desarrollo a medida para un control y personalización sin precedentes.

## 5.0 Opción 2: Desarrollo a Medida con Python

### 5.1 Arquitectura Técnica Propuesta

El desarrollo a medida representa la solución definitiva para ProAlto si el objetivo es obtener el máximo control sobre la seguridad, la personalización de la experiencia y la escalabilidad a largo plazo. Este enfoque permite construir una solución perfectamente adaptada a los procesos de negocio existentes y futuros, sin depender de las limitaciones de una plataforma de terceros. La arquitectura y el stack tecnológico recomendados para este desarrollo son los siguientes:

- **Lenguaje y Framework Backend:** Se recomienda **Python** junto con el microframework **Flask**. Esta combinación es ideal para construir aplicaciones backend ligeras, eficientes y escalables, y es ampliamente utilizada en el desarrollo de chatbots y APIs.
- **API de Mensajería:** Se utilizará la **API oficial de WhatsApp Business (Cloud API de Meta)**. La comunicación se gestionará a través de Webhooks, un mecanismo que permite a Meta notificar a nuestro servidor en tiempo real cada vez que se recibe un nuevo mensaje.
- **Base de Datos:** El backend en Python establecerá una **conexión directa y segura con la base de datos PostgreSQL** existente de ProAlto, permitiendo realizar consultas en tiempo real para obtener información de los créditos.
- **Procesamiento de Lenguaje Natural (Opcional pero Recomendado):** Para una comprensión más avanzada de las intenciones del usuario, se puede integrar la **API de OpenAI (GPT-4)**. Esto permitiría al bot interpretar solicitudes complejas formuladas en lenguaje natural, en lugar de depender únicamente de menús de botones.
- **Despliegue:** La aplicación se desplegará en servicios en la nube de alta disponibilidad como **Amazon Web Services (AWS), Google Cloud Platform (GCP) o Heroku**, garantizando un funcionamiento continuo y seguro. Esta arquitectura proporciona una base sólida y flexible para construir un asistente virtual robusto. A continuación, se detalla la guía de implementación técnica.

### 5.2 Guía Técnica de Implementación con Python y Flask

- **Configuración del Entorno de Meta for Developers:**
- Crear una nueva App en el portal de Meta for Developers.
- Configurar el producto "WhatsApp Business API" dentro de la App.
- Obtener un token de acceso temporal para pruebas y generar un token de acceso permanente para producción.
- Configurar el endpoint del Webhook, que será la URL pública de nuestro servidor Flask donde Meta enviará las notificaciones de mensajes.
- **Desarrollo del Servidor Backend con Flask:**

- Crear un nuevo proyecto de Python e instalar Flask.
- Desarrollar un servidor Flask con dos endpoints (rutas) principales:
- Un endpoint GET para la verificación del Webhook. Meta enviará una solicitud a esta ruta con un token de desafío (hub.challenge) que el servidor debe devolver para confirmar que es el propietario del endpoint.
- Un endpoint POST para recibir las notificaciones de mensajes entrantes. Cada vez que un usuario envíe un mensaje al bot, Meta enviará un payload JSON a esta ruta con el contenido del mensaje.
- **Implementación de la Lógica de Negocio:**
- Dentro del endpoint POST, escribir el código para analizar el payload JSON recibido de WhatsApp.
- Extraer la información clave: el número de teléfono del remitente (from), el contenido del mensaje (text, image, document, etc.) y el tipo de mensaje.
- **Integración Segura con PostgreSQL:**
- Utilizar una librería de Python como psycopg2 para conectar de forma segura a la base de datos PostgreSQL de ProAlto.
- Implementar una capa de autenticación del usuario. Antes de ejecutar cualquier consulta financiera (ej. SELECT saldo FROM credits WHERE cliente\_id = ?), el bot debe validar la identidad del usuario a través de un flujo conversacional (ej. solicitando los últimos dígitos de su cédula).
- Escribir funciones que ejecuten las consultas SQL necesarias para obtener saldos, estados de solicitud, etc., y devuelvan los resultados.
- **Mapeo de Intenciones y Respuestas:**
- Diseñar la lógica principal que, basado en el contenido del mensaje del usuario, determine qué acción tomar.
- Si el mensaje es "Consultar saldo", se activa el flujo de autenticación y la consulta a la base de datos.
- Si el usuario envía un documento, el código lo procesará y almacenará de forma segura.
- Una vez obtenida la información, se debe construir la respuesta y enviarla de vuelta al usuario utilizando la API de WhatsApp. Esto se hace realizando una solicitud POST al endpoint de mensajes de la API de Meta.
- **Gestión de la Seguridad y Autenticación:**
- Implementar un sistema de gestión de estado de la conversación para cada usuario (ej. usando un diccionario en memoria o una base de datos Redis). Esto permite que el bot recuerde el contexto, como por ejemplo, que está en medio de un proceso de validación de identidad.
- Asegurarse de que todas las credenciales (tokens de API, contraseñas de base de datos) se almacenen de forma segura como variables de entorno y no directamente en el código.
- **Despliegue de la Aplicación:**
- Desplegar la aplicación Flask en un servicio en la nube (ej. AWS Elastic Beanstalk, Heroku, Google App Engine).
- Asegurarse de que el endpoint del Webhook sea público y esté protegido con HTTPS.
- Configurar las variables de entorno en la plataforma de despliegue para que la aplicación tenga acceso a todas las credenciales necesarias. Independientemente de

la opción de desarrollo elegida, el cumplimiento de la normativa colombiana de protección de datos es un requisito ineludible.

## 6.0 Consideraciones de Cumplimiento Normativo (Colombia)

La implementación de un bot que maneja información financiera y personal de clientes en Colombia exige un cumplimiento estricto de la legislación sobre protección de datos personales. Dado que el bot gestionará datos sensibles como números de cédula, información de ingresos y detalles de créditos, adherirse a la Ley de Habeas Data es un pilar crítico para la operación, la confianza del cliente y la mitigación de riesgos legales. El diseño y la operación del bot de ProAlto deben cumplir, como mínimo, con los siguientes requisitos esenciales derivados de la **Ley 1581 de 2012** y la **Ley 2157 de 2021 ("Borrón y Cuenta Nueva")** :

- **Autorización Expresa:** Al inicio de la primera interacción con un nuevo usuario, el bot debe solicitar de manera clara y registrar el consentimiento explícito para el tratamiento de sus datos personales. El usuario debe aceptar activamente antes de que se recopile cualquier información.
- **Política de Tratamiento de Información (PTI):** El bot debe proporcionar un mecanismo sencillo (como un botón o un enlace) para que el usuario pueda consultar la Política de Tratamiento de Información de ProAlto en cualquier momento. Esta política debe detallar la finalidad del tratamiento de los datos y los derechos que le asisten al titular.
- **Seguridad y Confidencialidad:** La solución técnica implementada debe garantizar la seguridad de la información. Esto incluye el cifrado de la comunicación (provisto por la API oficial de WhatsApp) y el almacenamiento seguro de los datos personales y financieros en las bases de datos de ProAlto.
- **Derechos del Titular:** El bot debe ser un canal válido a través del cual los usuarios puedan ejercer sus derechos fundamentales de **conocer, actualizar, rectificar y suprimir** sus datos personales, así como revocar la autorización de tratamiento. El cumplimiento de estos principios no es opcional; es la base para construir una relación de confianza y transparencia con los clientes de ProAlto. Con estas consideraciones en mente, podemos trazar la hoja de ruta final para el proyecto.

## 7.0 Hoja de Ruta y Próximos Pasos

Tras analizar el contexto estratégico, las necesidades del cliente y las opciones técnicas, este informe concluye con una hoja de ruta estructurada para la ejecución del proyecto. La implementación se llevará a cabo por fases para garantizar un desarrollo controlado, una auditoría rigurosa y un lanzamiento exitoso que minimice los riesgos y maximice el impacto positivo. La hoja de ruta de implementación recomendada es la siguiente:

- **Fase 1: Diseño y Planificación (Semanas 1-2):**
  - Mapeo final y detallado de los flujos conversacionales para los cinco casos de uso prioritarios.
  - Definición del tono de voz y la personalidad del bot, asegurando que se alinee con la marca ProAlto (profesional, confiable y empático).
  - Selección final de la vía de desarrollo (No-Code vs. A Medida).
- **Fase 2: Configuración Técnica (Semanas 3-4):**
  - Verificación de la cuenta en Meta Business Manager y aprobación de la línea para la API de WhatsApp.



- Configuración inicial de la plataforma seleccionada (ManyChat o el entorno de desarrollo Python/Flask).
- Desarrollo y exposición segura de los endpoints de la API interna para la consulta a la base de datos PostgreSQL.
- **Fase 3: Desarrollo e Integración (Semanas 5-8):**
- Construcción de la lógica completa del bot, incluyendo todos los flujos, validaciones y la gestión de documentos.
- Programación y prueba de las integraciones con el sistema de ProAlto.
- Implementación de los protocolos de seguridad y validación de identidad.
- **Fase 4: Pruebas y Auditoría Legal (Semana 9):**
- Realización de pruebas funcionales y de estrés con un grupo de usuarios piloto para identificar y corregir errores.
- Auditoría exhaustiva de todos los flujos por parte del equipo legal para garantizar el cumplimiento irrestricto de la normativa de Habeas Data.
- **Fase 5: Lanzamiento y Monitoreo (Semana 10 en adelante):**
- Despliegue oficial del bot y comunicación a la base de clientes.
- Seguimiento continuo de métricas clave de rendimiento: tasa de resolución automática, tiempo promedio de respuesta, escalaciones a agentes humanos y encuestas de satisfacción del cliente (CSAT).
- Iteración y mejora continua de los flujos basada en los datos recopilados. La ejecución de este proyecto no solo optimizará los procesos internos de ProAlto, sino que también la posicionará como un líder en la transformación digital del sector de libranzas en Colombia. Al ofrecer un canal de servicio al cliente inteligente, seguro y disponible 24/7, ProAlto reforzará la confianza de sus clientes y sentará las bases para un crecimiento operativo sostenible y eficiente.