

Panel de control y reportaje de asistentes virtuales

Docente: Nicolás Cenzano.

Nombre: Vicente Alonso Llanos Gutiérrez.

Carrera: Ingeniería Civil Informática.

Fecha: Diciembre, 2023.

1. Resumen Ejecutivo

Wots es una empresa que utiliza las inteligencias artificiales generativas modernas para desarrollar soluciones de whatsapp, creando asistentes virtuales inteligentes capaces de generar conversaciones reales con clientes para recolectar los datos necesarios por las empresas que los utilicen. Esto permite automatizar múltiples áreas de las interacciones con los clientes, como la venta de productos, reservas de hoteles, confirmación de entregas de despacho y muchos más.

El éxito de Wots ha causado la necesidad de automatizar procesos internos en las relaciones con sus propios clientes. En particular, durante la pasantía desarrollada, se ha visto la necesidad de crear un sistema que permita automatizar el reportaje de la información generada por los asistentes virtuales. Actualmente esta información se entrega de forma mensual a los clientes de Wots, pero los reportes deben ser creados de forma manual cada mes, lo que significará un problema en el aspecto de la escalabilidad de la empresa en el futuro.

Buscando encontrar una solución a este problema, durante el período de pasantía se desarrolló un sitio web con un Dashboard orientado al cliente, que permitirá acceso en tiempo real a los datos de uso de los asistentes virtuales. Este producto permitirá establecer una plataforma propia de Wots sobre la que se podrán crear y agregar múltiples funcionalidades adicionales, facilitando el crecimiento de la empresa durante el año 2024.

Finalmente, el Dashboard fue desarrollando utilizando Firebase para la administración de las bases de datos y de usuarios, junto con todos los procesos de back end del sitio, mientras que el front end fue desarrollado en React.js. El producto permite controlar los usuarios y sus permisos por los administradores del sitio web, mientras que, una vez ingresados, los clientes pueden entrar al sitio, administrar su información personal de la cuenta y ver la información relevante de sus asistentes virtuales.

2. Abstract

Wots is a company leveraging generative AIs to craft WhatsApp solutions, constructing intelligent chatbots capable of engaging in real conversations with customers, and gathering the necessary data for the its client companies. This allows for the automation of various customer interactions, including product sales, hotel reservations, delivery confirmations and more.

Wots's success has led to the need to automate its own internal processes in dealing with its clients. Specifically, during the internship, it was identified the need to create a system that allows for automatization in the reporting process of information generated by Wots's chatbots. Currently, this information is delivered monthly to Wots's clients, but these reports have to be manually gathered and created each month, posing scalability challenges for the company.

In an effort to adress this issue, a website whit a client-oriented dashboard was developed the internship. This dashboard provides real-time access our chatbot's use data. This product Will establish a dedicated Wots Platform, on which multiple additional functionalities can be developed, facilitating the company's growth during the year 2024.

The Dashboard was developed using Firebase for users and database management, along with backend processes, and the frontend was developed in React.js. The producto allows website administrators to control all users and their permissions. Customers can Access the site and manage their account's personal information and view their chatbot's relevant information

Índice

01. Resumen Ejecutivo	2
02. Abstract	3
03. Contexto	5
04. Problema	6
05. Objetivos.....	7
06. Estado del Arte.	8
07. Solución y Metodologías.	14
08. Medidas de Desempeño.	17
09. Plan de implementación y desarrollo.	18
10. Análisis de Riesgo.	19
11. Evaluación Económica.....	23
12. Resultados.	25
13. Conclusiones.	27
14. Anexos.	28

3. Contexto.

Wots es una empresa fundada en 2022, ofrece soluciones a sus clientes haciendo uso de las tecnologías de inteligencia artificial emergentes de la actualidad. La empresa define un “WOT” como “Whatsapp Operational Task”, un Bot de alta tecnología que utiliza el canal de WhatsApp para solucionar una tarea operacional específica, o, en otras palabras, asistentes virtuales de whatsapp que utiliza las tecnologías de inteligencias artificiales modernas para automatizar tareas.

Actualmente, los asistentes virtuales de Wots tiene presencia en 4 países y en 10 industrias, ofreciendo soluciones en productos del sector público, en atención al cliente de empresas de automotriz, en medios de comunicación y noticias por WhatsApp y mucho más.¹

De esta forma, clientes de Wots están formados por otras empresas, las cuáles utilizan los asistentes virtuales y tecnologías ofrecidas para automatizar las interacciones con sus propios clientes, facilitando la realización de múltiples procesos.

Por otro lado, los asistentes virtuales son desarrollados utilizando el servicio “LandBot”, un software online constructor de chatbots. Este sitio permite los asistentes virtuales con facilidad, haciendo uso de diagramas de flujos de conversación por bloques, lo que lleva a una alta rapidez en la producción y entrega de nuevos asistentes virtuales. LandBot ofrece un registro y reporte de su uso y, adicionalmente, se utiliza Google sheets para almacenar información recolectada durante las conversaciones de los asistentes virtuales.

La pasantía se realizó en el área de desarrollo de software de la empresa, de esta forma resultó necesario realizar trabajos en los productos de los clientes de Wots, así como en el desarrollo del proyecto de pasantía.

1. <https://www.getwots.com/>

4. Problema.

Si bien, el desarrollo de los asistentes virtuales es sencillo y dinámico, la forma de operar de la empresa presenta grandes problemas que llevarán a una dificultad en su crecimiento a futuro, Wots debe trabajar con cada clientes de forma individual y personalizada. Esto quiere decir que todos los asistentes virtuales y documentos de almacenamiento de datos debe ser creado desde cero y entregado a cada cliente nuevo.

Uno de los aspectos que dejan en mayor evidencia este problema, está en la forma en la que se manejan y se entregan los reportes de uso y datos de los asistentes virtuales.

Parte de la información de uso de los chatbots se encuentra en Landbot, dónde únicamente Wots tiene acceso, mientras que otra porción de los datos es recolectada durante las conversaciones y se almacena automáticamente en archivos de Google sheets. Esta información debe ser recopilada, confeccionada y entregada manualmente a cada cliente de Wot. Este trabajo es realizado por el director de Wots de forma mensual, y ocupa un promedio de 2 hora en la creación de cada reporte por cliente.

Actualmente la empresa cuenta con 6 clientes activos para los que necesita estar generando informes mensuales, lo que corresponde a 12 horas mensuales dedicadas a su desarrollo. Si bien, en la actualidad no se trata de una pérdida de tiempo importante, al tratarse de una relación lineal entre número de clientes y tiempo dedicado mensualmente, el crecimiento de la empresa se verá seriamente limitado por este método de creación de reportes.

$$\text{Horas Mensuales} = N^{\circ} \text{ de clientes}$$

$$\text{Horas Mensuales} = 12 \text{ hrs.}$$

Fórmula 1: Métricas del Problema

5. Objetivos.

Para solucionar el problema, se cumplirá el objetivo SMART siguiente:

El objetivo general del proyecto será entregar un servicio que ofrezca a los clientes la información relevante del uso de sus asistentes virtuales de forma automática, para reducir el tiempo dedicado a la confección de reportes.

Los objetivos específicos del proyecto serán los siguientes:

1. Crear una Plataforma de la empresa Wots que entregue la información relevante del uso de los asistentes virtuales y que pueda ser utilizada por administradores y clientes.
2. Almacenar en un solo lugar la información relevante de todos los clientes de Wots.
3. Reducir y centralizar el tiempo dedicado a la confección de reportes en, en vez de dedicar tiempo a cada reporte por separado, solo será necesaria la mantención del sitio web.
4. Ofrecer sistemas de seguridad de usuarios y manejo de datos.
5. El Sistema deberá tener capacidad de crecimiento, para automatizar más funcionalidades en el futuro.

6. Estado del Arte.

- Servicios Actuales: Google Sheets y Landbot.

Actualmente Wots utiliza Google Sheets para almacenar los datos de uso de sus asistentes virtuales, mientras que se utiliza el sitio web LandBot para la creación de los asistentes virtuales. Una ventaja del servicio Landbot es que permite almacenar los datos recolectados durante una interacción con su asistente virtual con facilidad.

Una posible solución para el problema de reportaje automático es hacer uso de los sistemas que ya se utilizan en la empresa. Google Sheets se puede integrar con Google Scripts para automatizar procesos, como la recolección y reportaje de datos a un correo electrónico.

Por otro lado, Landbot ofrece acceso a las métricas de uso de sus asistentes virtuales, sin embargo, no ofrece un medio para exportar esta información, ni una API que permita obtenerla de forma externa, por lo que el acceso a esta información de uso queda restringida únicamente a los usuarios de Wots, los dueños y desarrolladores de los asistentes virtuales.

- Diseño y visualización de Datos:

Es importante tener en consideración la importancia del diseño de la visualización de los datos, “El propósito del diseñador en el diseño de una visualización de datos es crear un entregable que sea bien recibido y fácil de entender por el lector. Todas las elecciones de diseño y las implementaciones particulares deben cumplir este propósito.”²

De esta forma, una opción poderosa para el desarrollo de un sistema de reportería, es el desarrollo de un dashboard o panel de control dirigido a los clientes. Los dashboards ofrecen la posibilidad de diseñar la visualización de datos a través de gráficos e imágenes y tienen como objetivo principal hacer que la información compleja sea accesible y fácil de digerir.³

2. Iliinsky, N., Steele, J. (2011), Data Visualizations. O'Reilly

3. Subotin, S. Diseño de Dashboard – Consideraciones y Mejores Prácticas

<https://www.toptal.com/designers/data-visualization/disenio-de-dashboard-consideraciones-y-mejores-practicas>

- Dashboard:

Los pasos necesarios para desarrollar un dashboard pueden variar de proyecto a proyecto, según las necesidades y objetivos. De esta forma, los aspectos técnicos relevantes para resolver el problema que se está tratando durante esta pasantía consisten en la adquisición de los datos relevantes, el diseño del front end, y la creación de un sistema de alimentación que actualice la data del dashboard de forma periódica.⁴

Continuando lo dicho anteriormente, se optó por separar las opciones de servicios en 3 áreas generales. Por un lado, se evaluarán frameworks que se puedan utilizar para diseñar y desarrollar el front end del sitio web, luego, se verán opciones de sistemas de back end, que puedan alimentar y actualizar la data del sitio web, y, por último, servicios de bases de datos, donde se almacenará información de usuario, así como datos de uso de asistentes virtuales.

Los criterios que se utilizaron para seleccionar los frameworks y servicios que se verán a continuación son dos. Por un lado, se revisaron sistemas que ocuparan lenguajes de programación que en los que se tuviera experiencia, para evitar la necesidad de añadir el tiempo de aprendizaje de un nuevo lenguaje de programación al desarrollo del proyecto. Estos lenguajes fueron Javascript y Django.

Por otro lado, se seleccionaron frameworks por su popularidad, al tratarse de ser servicios ampliamente utilizados, existiría una mayor comunidad de programadores en la que apoyarse durante en desarrollo de las funcionalidades.⁵

4. Chisel Analytics (2020), Building a Dashboard: Framing the problem and getting started
<https://chiselanalytics.com/building-a-dashboard-framing-the-problem-and-getting-started/>

- FrontEnd.⁵

- Angular.^{6,7}

Angular es un Framework de FrontEnd lanzado el 2016 por Google, Angular permite la creación de aplicaciones web dinámicas. Es un framework basado en TypeScript y se utiliza para aplicaciones de una sola página.

Al ser un servicio ofrecido por Google, Angular cuenta con una documentación robusta y ofrece la seguridad de ser un servicio que será mantenido a largo tiempo.

- React.^{8,7}

React es un Framework de FrontEnd de código abierto, utilizado para la creación de interfaces de usuario basadas en componentes. Fue desarrollado por Facebook en 2013 y es actualmente mantenido tanto por Facebook como por la comunidad de Código Abierto.

React utiliza JavaScript y permite la creación de aplicación web y aplicación móvil utilizando la misma base de código, gracias al Framework de React Native.

React es de los framework más utilizados en el mundo, lo que significa una gran cantidad de documentación y soporte comunitario.

- Vue.js.^{9,7}

Vue es un Framework de código abierto lanzado el 2014. Es mayormente utilizado para la creación de aplicaciones de una sola página y es considerado uno de los Frameworks con la curva de aprendizaje más simple. Sin embargo, no cuenta con el respaldo de Google o Facebook con el que cuentan React o Angular.

5. Se utilizó Knowledge hut para evaluar la popularidad de frameworks de frontend:

<https://www.knowledgehut.com/blog/web-development/front-end-development-frameworks>

6. <https://angular.io/docs>

7. Joshi, M. (2023), Angular vs React vs Vue: Core Differences.

<https://www.browserstack.com/guide/angular-vs-react-vs-vue>

8. <https://react.dev/>

9. <https://vuejs.org/>

- BackEnd.¹⁰.

- Express JS.¹¹.

Express JS es un Framework de Backend para Node.js, que utiliza JavaScript. Es un framework cuya fortaleza está en su simplicidad de uso. Tiene un enfoque modular, lo que permite hacer adiciones a la aplicación a medida surjan las necesidades.

Este Framework cuenta con un robusto soporte comunitario tanto en librerías y middleware, como en ayuda y guías de uso.

- Django.¹².

Django es un Framework de programación desarrollado en Python que ofrece una gran variedad de herramientas y funcionalidades. Su principal objetivo es permitir construir aplicaciones web complejas de forma rápida.

Django permite flexibilidad y facilidad en integración de bases de datos para el desarrollo de sitios web, ofreciendo un sistema de ORM (Mapeo Relacional de Objetos).

10. Nguyen, T. (2023), Express JS vs Django: A comprehensive Comparison of two leading web frameworks.
<https://www.frontendmag.com/insights/express-js-vs-django/>

11. <https://expressjs.com/>

12. <https://www.djangoproject.com/>

- Bases de Datos y Cloud Services.

- MySQL¹³.

MySQL es un sistema de gestión de Bases de Datos desarrollado por Oracle y es la base de datos de código abierto más popular del mundo.

MySQL maneja bases de datos de tipo relacionales, lo que ofrece alta rapidez en la lectura de datos, convirtiéndose en una buena opción para el desarrollo con bases de datos de alta complejidad, y con una gran cantidad de datos. Se utilizan sql queries para realizar trabajos de lectura y escritura de datos.

MySQL puede ser integrado tanto con Django como con Express Js, convirtiéndola en una buena opción para el desarrollo del sitio web.

- AWS.^{14, 15}

AWS, o Amazon Web Services, es una plataforma de cloud computing que ofrece sistemas de almacenamiento para bases de datos de tipo relacional, como no relacional. Adicionalmente, AWS ofrece una gran cantidad de servicios para el desarrollo de aplicaciones web, permitiendo manejar las tareas del BackEnd del sitio web.

AWS ofrece sistemas de PaaS (Platform as a Service), IaaS (Infrastructure as a Service) y SaaS (Software as a Service), permitiendo ofrecer soluciones para cualquier necesidad que surja durante el desarrollo de aplicaciones web.

Los sistemas de Almacenamiento de AWS cuentan con un alto nivel de seguridad, siguiendo estándares internacionales en la seguridad de datos sin necesidad de pago adicional.

13. <https://dev.mysql.com/doc/>

14. <https://aws.amazon.com/es/>

15. Todos los datos fueron obtenidos de “Firebase vs AWS vs Azure – What are the differences?”
Clark, J. (2022) Firebase vs AWS vs Azure – What are the differences?.
<https://blog.back4app.com/firebase-vs-aws-vs-azure/>

AWS no cuenta con un precio base para sus servicios, ofrece una gran cantidad de servicios base gratis, y los servicios que deben ser pagados, se calcula su costo a medida se utilice con el AWS Pricing Calculator, de forma que, si un servicio no es utilizado, no generará costos adicionales en la empresa

- Firebase.^{16, 15.}

Firebase es un servicio de tipo BaaS (Backend as a Service) publicado en 2011 y adquirido por Google en 2014.

Firebase ofrece manejo de Bases de Datos no relaciones, o bases de datos NoSQL, ideal para bases de datos sin alta complejidad, lo que permite una gran escalabilidad.

Al ser Firebase un producto de Google, ofrece una gran facilidad en la implementación de sistemas de autenticación, utilizando las APIs de Gmail, Facebook, Github, Twitter, etc...

Firebase ofrece un sistema de pago similar al de AWS, Pay-As-You-Go tiene costos por uso, por ejemplo, tiene costo de \$0.01/verificación y \$0.026/GB de almacenamiento.

- Azure.^{17, 15.}

Microsoft Azure es un sistema de cloud computing similar a AWS, que ofrece servicios de IaaS PaaS y SaaS. Siendo un sistema popular que cuenta con la confianza de grandes empresas como MasterCard, LinkedIn, GitLab y GoDaddy.

Ofrece una gran variedad de tipos de bases de datos, como NoSQL, relacional, llave-valor, etc...

Azure ofrece modelos de pago similares a AWS y Firebase, ofreciendo una gran gama de servicios totalmente gratis, mientras que para proyectos con mayores necesidades pueden optar por un modelo de Pay-As-You-Go.

15. Todos los datos fueron obtenidos de “Firebase vs AWS vs Azure – What are the differences?”

Clark, J. (2022) Firebase vs AWS vs Azure – What are the differences?.

<https://blog.back4app.com/firebase-vs-aws-vs-azure/>

16. <https://firebase.google.com/>

17. <https://azure.microsoft.com/>

7. Solución y Metodologías.

Para la solución final del problema presentado, se optó por desarrollar un dashboard que permita a los clientes de Wots visualizar los datos de uso de sus asistentes virtuales, ya que esta opción, además de resolver el problema el reportaje, ofrece un producto en el que se podrá continuar trabajando y expandiendo para resolver otros problemas de crecimiento que tenga actualmente la empresa.

Para el desarrollo del producto se decidió utilizar la metodología de WaterFall. Esta es una metodología que consiste en el desarrollo secuencial de tareas para el desarrollo y entrega de un producto, estas etapas serán: Levantamiento de requisitos, diseño, desarrollo, testeo, despliegue y mantenimiento del sitio.

Si bien la metodología de WaterFall presenta mayor inflexibilidad que otras metodologías, (por ejemplo, las metodologías ágiles), se consideró adecuada para el desarrollo del proyecto, ya que el levantamiento de un producto de software de panel de control presenta la suficiente documentación y ejemplos de desarrollo para utilizar un desarrollo secuencial, en vez de un desarrollo iterativo. Esto permitirá una planificación que entregará el producto en las fechas necesarias.

Los requerimientos que debe tener el dashboard y con los que debe cumplir la solución entregada son los siguientes:

- Sistema de Autenticación e Inicio de Sesión.

El Sitio debe permitir que cada usuario o cliente acceda a su propio panel de control de forma segura. Los datos de los usuarios deben ser almacenados de forma segura para evitar vulneraciones de seguridad.

- Sistema de permisos para los usuarios.

Debe existir un sistema de permisos que permita la existencia de un usuario administrador, que pueda manejar y crear las cuentas de los clientes nuevos, para evitar la creación de usuarios que no se vayan a utilizar.

- Acceso y visualización de los datos de cada usuario.

Cada usuario debe poder acceder a los datos específicos de sus asistentes virtuales y estos se deben presentar de forma simple y de fácil comprensión.

- Estructura flexible y que permita el crecimiento del sitio.

La estructura del sitio debe tener la flexibilidad para presentar los distintos tipos de datos que cada usuario necesite. Al mismo tiempo, esta flexibilidad debe permitir la integración de nuevos sistemas adicionales al reportaje (por ejemplo, acceso a tienda de servicios de Wots), de forma que, en el futuro, el Dashboard ofrezca mayor utilidad a sus usuarios.

Para el desarrollo de la solución, se utilizará Google Sheets y Firebase para el almacenamiento de datos del dashboard y el manejo de procesos de backend. Firebase presenta la ventaja significativa de ofrecer sistemas de autorización y autenticación con gran facilidad de implementación, mientras que su base de datos no relacional resulta ideal para los datos que se manejan por los clientes, donde cada archivo de reporte tiene una estructura única. Por otro lado, se decidió utilizar Google Sheets debido a que ya se almacena toda la información en sus documentos y es el único servicio que tiene una buena integración con Landbot para el almacenamiento de los datos de las conversaciones.

De esta forma, se utilizará Google Sheets y Firebase en conjunto para hacer el manejo del backend, se recopilarán los datos de los documentos y se subirán a la base de datos de Firebase, Firestore Database, donde podrán ser manejado por el Dashboard.

Además, se utilizará firebase para el manejo de los usuarios, se utilizará su servicio de autenticación para administrar y autenticar a los clientes y administradores de Wots, y se almacenará la información de los usuarios en la base de datos Firestore.

Por último, debido a las necesidades del sitio web a desarrollar, se decidió utilizar React para el desarrollo del front end del sitio web. React permite la creación de un sitio web de múltiples páginas con mayor facilidad que las otras opciones, mientras que react native permitirá expandir alcance del proyecto en el futuro, permitiendo crear una versión móvil con facilidad.

A continuación, se presenta un diagrama del proceso de la solución. Los asistentes virtuales de Landbot recopilan la información de las conversaciones y las suben un documento de Google sheets. Luego, se utiliza firebase functions para recolectar la información nueva y agregarla a la base de datos Firestore, donde estará almacenada la información de reporte de las empresas. Por otro lado, las páginas del sitio web desarrolladas en react se conectarán con los sistemas de autenticación de firebase para permitir el ingreso de sus usuarios y, según la información y permisos del usuario, se presentarán los datos del asistente virtual que corresponda.

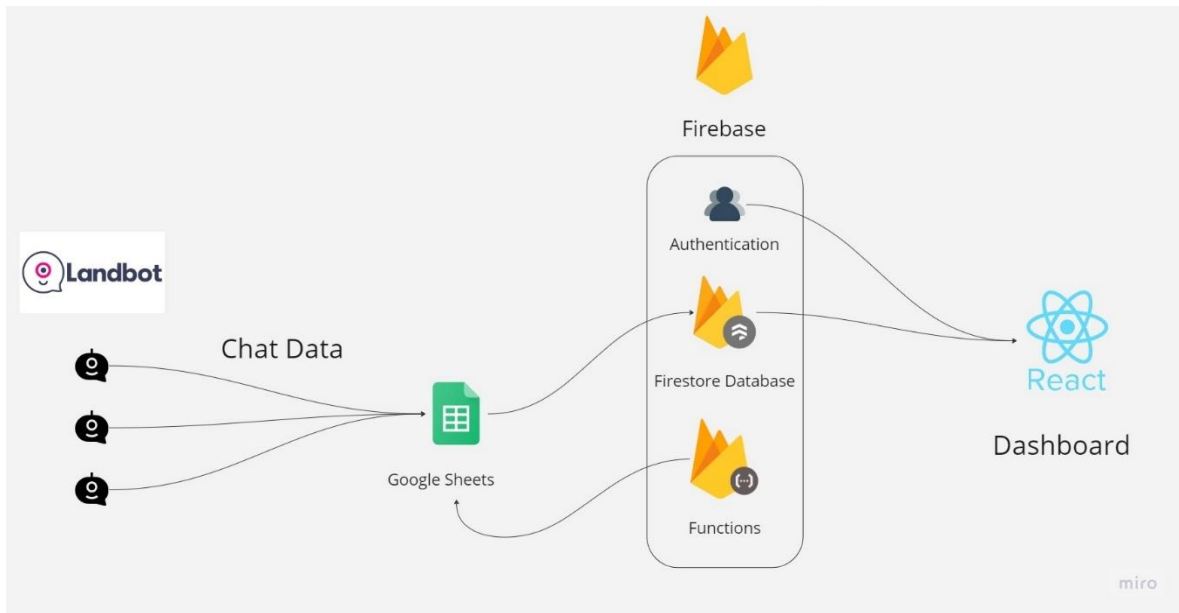


Ilustración 1: Diagrama de flujo de Dashboard.

8. Medidas de Desempeño.

Para evaluar el éxito del proyecto se medirán las siguientes medidas de desempeño:

1. Engagement:

Se medirá el uso de la plataforma por los clientes para evidenciar su uso y utilidad, y se analizará el porcentaje de cliente que utilizan el sistema de reportaje.

$$\frac{N^{\circ} \text{ de clientes que utilizan el sitio}}{N^{\circ} \text{ de clientes de Wots}} = KPI 1$$

Fórmula 2: Medida de Desempeño 1

2. Tiempo:

Se medirá el tiempo dedicado en la mantención y programación del sitio web en los aspectos dedicados a la entrega de reportes, para poder ser comparado con el tiempo que se dedicaba previamente.

$$\frac{\text{Horas Persona}}{N^{\circ} \text{ de clientes de wots}} = KPI 2$$

Fórmula 3: Medida de desempeño 2

3. Seguridad:

Para evaluar la seguridad del sitio, se medirán todas las brechas y riesgos de seguridad del sitio de forma mensual.

$$\text{Brechas de seguridad mensual} = KPI 3$$

Fórmula 4: Medida de desempeño 3

4. Almacenamiento:

Por último, los datos de uso de los asistentes virtuales se encuentran en múltiples documentos separados de Google sheets, se medirá que porcentaje de estos datos han sido integrados exitosamente en la base de datos del sitio web.

$$\frac{\text{Datos en Base de datos}}{\text{Datos en Google Docs}} = KPI 4$$

Fórmula 5: Medida de desempeño 4

9. Plan de implementación y desarrollo.

La implementación del proyecto se separó en 3 etapas principales, la etapa de Planificación y Formulación de Proyecto, la etapa de Prototipado y la etapa de desarrollo del Dashboard.

La etapa de Planificación y Formulación comenzó en agosto y se finalizó la tercera semana de septiembre, en esta etapa se trabajará en la Identificación de la problemática, la Formulación del Proyecto y en el Estudio del estado del Arte.

En la etapa de Prototipado se trabajarán en 2 prototipos del dashboard para recibir retroalimentación sobre la funcionalidad del sitio web antes de comenzar el trabajo en el sitio definitivo. Esta etapa comenzó en octubre y se finalizó la primera semana de noviembre.

Finalmente, se dedicó noviembre y diciembre al desarrollo final del sitio web, en esta etapa se trabajó en el código del sitio y, adicionalmente, fue necesario hacer el levantamiento de la base de datos y la creación de la cuenta de los clientes, y, finalmente, se mantendrá un constante período de pruebas.

Luego, a futuro, se mantendrá un constante análisis en los resultados del proyecto y se evaluará la posibilidad de implementar funcionalidades adicionales al sitio web.

Cabe mencionar que en el futuro existirá una cuarta etapa constante una vez se haya entregado el producto final. Debido a la naturaleza del dashboard que se entregó, se podrán implementar y desarrollar nuevas soluciones sobre el mismo sitio web. Gracias a esto, se utilizará la metodología ágil de SCRUM en el desarrollo de nuevas soluciones.

La Carta Gantt que se utilizó para guiar el desarrollo del proyecto se puede ver en el Anexo 1.

10. Análisis de Riesgo.

Se consideraron 6 principales riesgos presentes durante el desarrollo del proyecto y para su implementación exitosa. La matriz completa se puede encontrar en el punto 2 de la sección de anexos.

- Riesgo de proyección de trabajo.

Es posible que el trabajo requiera más trabajo del proyectado, esto tendría como consecuencia que el proyecto no se realice en el plazo estipulado.

La probabilidad de este riesgo se considera alta con un impacto medio, para mitigar el riesgo se utilizarán herramientas de aprendizaje rápido y de fácil uso.

- Servicios de Landbot

Landbot es un servicio que no ofrece un acceso por medio de APIs u otras alternativas para exportar los datos con la información de los asistentes virtuales. Como consecuencia, no sería posible mostrar los datos recolectados por Landbot (cantidad de mensajes enviados, conversaciones activas y cerradas).

Para mitigar este riesgo, el dashboard se manejará con los datos que se almacenan por separado en los documentos de Google Sheets

- Uso de Dashboard

Existe el riesgo de que los clientes no utilicen el Dashboard creado. Lo que tendría como consecuencia la necesidad de continuar confeccionando y entregando los reportes de forma manual, esto resultaría en un fracaso del proyecto.

Para evitar este riesgo, se fomentará el uso del Dashboard a los clientes constantemente, y se implementará en el futuro sistemas de reportaje automático que se envíen a los correos de los usuarios inscritos.

- Seguridad

Los riesgos de seguridad del sitio web son los más complicados de abordar y solucionar. La complejidad de los sitios crea múltiples oportunidades para que se presenten vulnerabilidades de los datos, servidores y privacidad de los usuarios. De esta forma, se analizarán el top 10 de vulnerabilidades de seguridad de OWASP.

La “Open web Application Security Project” (OWASP) es una organización sin fines de lucro con la misión de mejorar la seguridad de las aplicaciones web. “OWASP ha desarrollado una serie de recursos que describen las vulnerabilidades más comunes que existen en varios sistemas, incluyendo aplicaciones web (...). El más famoso de estos es el Top Ten de OWASP, que describen las diez vulnerabilidades más comunes”¹⁸.

Estas vulnerabilidades son: Control de acceso roto, fallas criptográficas, inyección, diseño inseguro, mala configuración de seguridad, componentes vulnerables y obsoletos, fallos de identificación y autenticación, fallas de integridad de datos y software, fallos de registro y monitoreo de seguridad y falsificación de solicitudes del lado del servidor. A continuación, se separará la forma en que se abordarán los riesgos en tres áreas, frontend, backend y las interacciones entre estos sistemas.

- Seguridad – FrontEnd (React):

En esta área se abordan los riesgos de diseño inseguro, componentes vulnerables y obsoletos, y, en menor parte, los aspectos de identificación y autenticación, control de acceso roto y falsificación de solicitudes. En el uso de javascript se presentan vulnerabilidades específicas del lenguaje “Las vulnerabilidades (...) aunque se produzca en el lado del cliente, finalmente se pueden convertir en amenazas del lado del servidor ya que una vez explotadas, permiten lanzar peticiones contra APIS en las que en un principio no se tiene autorización.”¹⁹

18. “Las 10 principales vulnerabilidades de seguridad de aplicaciones web de OWASP”
<https://www.checkpoint.com/es/cyber-hub/cloud-security/what-is-application-security-appsec/owasp-top-10-vulnerabilities/>

19. Rodríguez, D. R. (2022), Análisis de Servicios y Mecanismos de Seguridad desde el Diseño en front-ends de desarrollo Web.
<https://uvadoc.uva.es/bitstream/handle/10324/57335/TFG-G5843.pdf?sequence=1>

Para el desarrollo del front end del sitio web, se siguieron las prácticas listadas por el paper “Análisis de Servicios y Mecanismos de Seguridad desde el Diseño en front-ends de desarrollo Web” (2022, D. Raúl Rodríguez). Estas prácticas incluyen, pero no están limitadas a:

- Habilitar cifrado mediante TLS/SSL.
- Uso adecuado de Tokens de Seguridad.
- Utilizar herramientas de escaneo.
- Establecimiento de cookies seguras.
- Utilizar métodos de manipulación del DOM seguros.

De todas formas, con las medidas tomadas para la prevención de riesgos en el diseño del frontend, continúan existiendo nuevos riesgos que puedan surgir producto de vulnerabilidades nuevas en las librerías y servicios utilizados, por lo que, adicionalmente, se evaluará mensualmente el estado de las librerías y métodos utilizados en caso de ser necesario realizar actualizaciones.

- Seguridad – Backend:

El uso del servicio cloud de Firebase ofrece el uso de herramientas que permite proteger los datos y funciones que se utilizan. Para asegurar este servicio se utilizarán reglas de seguridad y prácticas de uso conforme a las pautas establecidas por firebase en el uso de sus servicios.²⁰

Adicionalmente, se utilizará el servicio Firebase Crashlytics para monitorear el estado del sitio web y los errores que puedan surgir en tiempo real. Este servicio nos permite identificar el problema de raíz y cuenta con un sistema de alertas en tiempo real de los nuevos errores, lo que nos permite llevar una trazabilidad de estos.¹⁹

20. Lista de tareas de seguridad de Firebase.

<https://firebase.google.com/support/guides/security-checklist?hl=es-419>

19. Rodríguez, D. R. (2022), Análisis de Servicios y Mecanismos de Seguridad desde el Diseño en front-ends de desarrollo Web.

<https://uvadoc.uva.es/bitstream/handle/10324/57335/TFG-G5843.pdf?sequence=1>

- Seguridad – Interacciones:

Por último, se protegerán las interacciones entre los servicios de front end y back end a través del uso de Firebase Functions. Este se trata de un servicio de Firebase que utiliza *serverless functions* para crear APIs y puede mediar las interacciones entre los componentes de Firebase y las aplicaciones que los utilicen y es la manera ideal de proteger la información sensible, como las llaves de acceso a firebase.¹⁹.

Se incluye en el anexo 2 la matriz de riesgos desarrollada con el resumen de los riesgos analizados y sus mitigaciones.

19. Rodríguez, D. R. (2022), Análisis de Servicios y Mecanismos de Seguridad desde el Diseño en front-ends de desarrollo Web.

<https://uvadoc.uva.es/bitstream/handle/10324/57335/TFG-G5843.pdf?sequence=1>

11. Evaluación Económica.

Para la evaluación económica del proyecto, se considerará encontrarse implementado en diciembre, por lo que se realizará una proyección del valor que generará a la empresa durante el año 2024. Existirán beneficios adicionales que genere la existencia de un Dashboard para la empresa que no se considerarán a continuación, ya que estos beneficios dependerán de la implementación de nuevos proyectos de desarrollo de funcionalidades.

Actualmente, en promedio, la empresa adquiere 1 cliente nuevo cada 2 meses y pierde un cliente cada 6 meses.

La implementación de un dashboard orientado al cliente tendrá un impacto en dos áreas fundamentales del negocio de Wots, en la adquisición de clientes nuevos y en la retención de los mismos. Esto se debe a que el sitio web ofrecerá la confianza y transparencia de los datos que maneja Wots.

Es por esta razón que, gracias al proyecto, y debido al tiempo que se ahorrará en la creación de los reportes y la consistencia con la que se podrá hacer entrega de estos, se espera tener un aumento de un 30% en la adquisición de clientes (1,3 clientes cada 2 meses) y un aumento en la retención de clientes de un 20% (Se pierde un cliente cada 7 meses). Estos valores son actualmente altos debido a la pequeña cantidad de clientes con los que cuenta la empresa. Este crecimiento se considerará constante durante el año 2024.

Se considerará que cada cliente significa un valor generado por la empresa de \$1.000.000 mensuales. Por motivos de privacidad de la empresa, este valor se trata de una representación ficticia de los ingresos, que servirá para visualizar el aporte que generará el proyecto.

Para los gastos en el uso de Firebase, debido a la pequeña cantidad de datos que se manejarán actualmente, se considerará que cada cliente significará un costo adicional en la suscripción de firebase de un máximo de 20 dólares al mes. El valor real será probablemente menor, pero se utilizará esta estimación para la evaluación económica. Se considerará el valor del dólar 900 pesos durante el año 2024.

En último lugar, se consideró una tasa de descuento ya establecida por Wots para el desarrollo e implementación de sus proyectos de Software, el valor de la tasa de descuento utilizada es de 12%.

Utilizando estos datos, se obtuvieron los siguientes valores de las estimaciones de VAN y TIR:

$$\$2.679.214 = VAN$$

$$38\% = TIR$$

Luego, se considerará como caso optimista uno donde la adquisición de clientes aumenta en un 50% y la retención de clientes se aumenta a 50% durante el primer año de implementación.

$$\$26.812.000 = VAN$$

$$55\% = TIR$$

Por último, se considerará uno pesimista donde no se producen cambios en la adquisición ni retención de clientes, pero los costos de firebase se mantienen. En este caso no se incluye un valor TIR dado que el proyecto no genera valor durante el período de tiempo.

$$\$ - 684.000 = VAN$$

Con los datos presentados, se puede concluir que, dado el costo de la inversión que se debe hacer para implementar el proyecto desarrollado, la propuesta resultará beneficiosa para la empresa. El motivo por el que no existen costos adicionales para su implementación, como, por ejemplo, el valor del dominio o de hosting, es debido a que la empresa ya tiene pagado el dominio y es un costo que va a existir independiente de si se implementa el proyecto o no, mientras que el hosting del sitio es realizado por una empresa diferente con la que se tiene un convenio, de forma que se puede levantar el sitio sin costo adicional.

Los flujos de caja desarrollado para realizar las evaluaciones económicas se pueden encontrar en los anexos 3, 4, 5 y 6.

12. Resultados.

El proyecto se entregará en diciembre de 2023, por lo que a continuación se proyectará los resultados esperados de su implementación durante el año 2024 sobre los KPIs establecidos previamente.

- KPI 1: Engagement

Se espera que inicialmente un 80% de los clientes utilicen la página por lo menos 1 vez de forma mensual para el acceso a sus reportes. Se espera que, fomentando el uso del sitio, este porcentaje aumente hasta un 100% para finales del año.

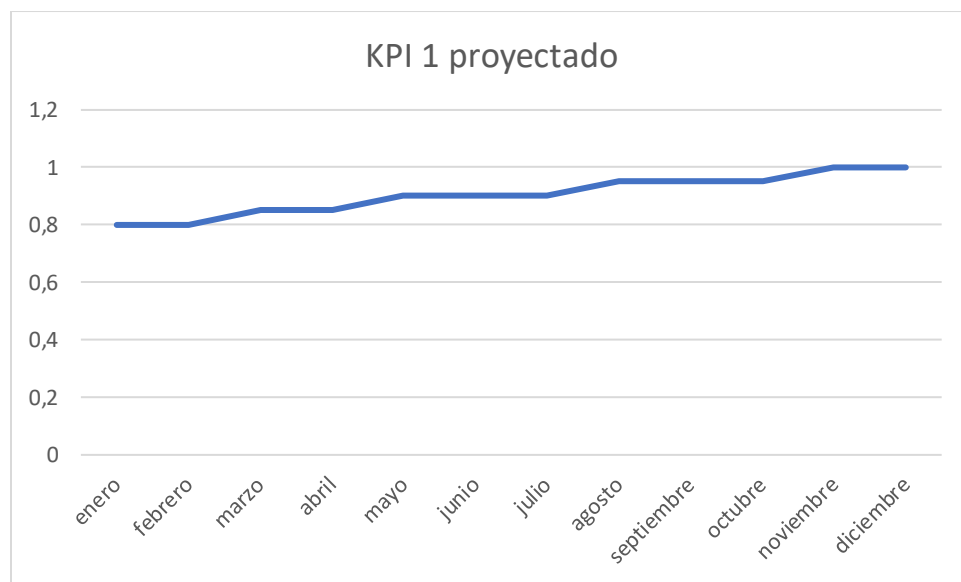


Gráfico 1: KPI 1 durante 2024

- KPI 2: Tiempo:

La finalización del proyecto eliminará el tiempo de confección de reportes, sin embargo, este tiempo pasará a ser el tiempo dedicado al desarrollo y mantención del sitio web. Se estima que con la integración de cada cliente nuevo será necesario dedicar 2 día de trabajo (16 horas) para preparar sus datos para ser integrados por el dashboard, con la adquisición de clientes actual de 1 cliente nuevo cada 2 meses, el promedio será de 8 horas mensuales dedicado a la confección y entrega de reportes a los clientes.

$$8 \text{ hrs} = \text{KPI 2}$$

Fórmula 6: Medida de desempeño 2 Proyectada.

- KPI 3: Seguridad:

Dado lo impredecible y variable que resulta la revisión y registro de vulnerabilidades de seguridad, se decide no realizar una proyección. De todas formas, cabe mencionar que será necesario mantener un registro y revisión constante para garantizar la seguridad del sitio web. En el momento de su lanzamiento, el Dashboard no presenta vulnerabilidades de seguridad identificadas.

- KPI 4: Almacenamiento:

Dada la facilidad de la integración entre Firestore y Google Docs, se podrá almacenar todos los datos presentes en los archivos de Google Docs en la base de datos de Firebase.

$$100\% = KPI\ 4$$

Fórmula 7: Medida de desempeño 4 Proyectada.

13. Conclusiones.

Tras la entrega del producto, cabe mencionar que el éxito del proyecto depende en gran medida de lo efectivo que sea Wots en el constante incentivo a los clientes de darle uso al dashboard creado y mantener una constante mantención del sitio, de forma que continúe siendo un producto estable y amigable para los clientes.

Por otro lado, un aspecto importante del proyecto realizado, es el abanico de oportunidades que genera para el desarrollo de nuevos proyectos contenidos en la plataforma creada. Tras la implementación del panel de control, se podrá desarrollar nuevos proyectos utilizando metodologías ágiles, lo que permitirá el desarrollo y entrega de nuevas funcionalidades, como, por ejemplo, implementar nuevas métricas medidas por los asistentes virtuales que se integren automáticamente al sitio web. También será posible integrar una tienda virtual en la página, lo que, integrado apropiadamente, podrá ofrecer mayor rapidez en la venta de nuevas funcionalidades para los clientes.

De esta forma, la entrega del producto no solo tendrá efectos positivos sobre las medidas de desempeño específicas al proyecto, sino que también ofrecerá facilidad para la creación de nuevos proyectos.

14. Anexos

Anexo 1: Carta Gantt

Tareas	Agosto					Septiembre				Octubre				Noviembre					Diciembre			
	S1	S2	S3	S4	S5	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S5	S1	S2	S3	S4
Planificación y Formulación																						
Identificación de problemática																						
Formulación de Proyecto																						
Estudio del estado del Arte																						
Prototipo																						
Desarrollo Prototipo 1																						
Entrega Prototipo 1																						
Desarrollo Prototipo 2																						
Entrega Prototipo 2																						
Dashboard																						
Desarrollo de Sitio Web																						
Levantamiento de Base de Datos																						
Conexión con Base de Datos																						
Creación de Cuentas de Clientes																						
Entrega de Producto																						
Período de pruebas																						

Figura 2: Carta Gantt



Anexo 2: Matriz de Riesgo.

Tipo de Riesgo	Riesgo		Impacto	Probabilidad	Evaluación		Respuesta
	Fuente	Consecuencia			Valor (1 a 9)	Nivel	
Experiencia	El proyecto requiere más trabajo del proyectado	El proyecto no se realizaría a tiempo	Medio	Alto	8	Alto	Se utilizarán herramientas de aprendizaje rápido y fácil uso
Seguridad	Los Frameworks de front end utilizados presentan vulnerabilidades de seguridad	Los datos de los clientes se encuentran expuestos a robos y alteraciones	Alto	Alto	9	Alto	Se harán revisiones cada mes de las librerías y servicios de Front end para mantenerlos actualizados a su última versión
Seguridad	Firebase no ofrece las medidas de seguridad necesarias en la gestión de usuarios y base de datos	Puede afectar la seguridad de los clientes y haría necesario utilizar otro Cloud Service para la implementación del proyecto	Alto	Bajo	5	Medio	Se utilizará AWS en caso de que Firebase no cumpla con las medidas de seguridad necesarias.
Seguridad	Las conexiones entre frontend y backend presentan posibles puertas de ataque	Los datos de los clientes se encuentran expuestos a robos y alteraciones	Alto	Medio	7	Medio	Todos los accesos de escritura y lectura de la base de datos serán mediados por cloud functions, para mediar el acceso de los datos
Técnico	LandBot no ofrece una API para el acceso al reporte de sus asistentes virtuales	No se pueden mostrar los datos de la API de Landbot en el Dashboard	Bajo	Alto	4	Bajo	En caso de ser así, el dashboard se manejará con los datos que se almacenan por separado en las google sheets.
Gestión	Los clientes no utilizan el Dashboard creado	Continúa siendo necesaria la creación de Reportes de forma manual y aumenta el tiempo de trabajo perdido	Medio	Alto	6	Medio	Se fomentará el uso del Dashboard a los clientes constantemente con sistemas de reportaje automático que se envíen a los correos de los usuarios inscritos.

Figura 3: Matriz de Riesgo

Anexo 3: Flujo de caja sin Proyecto.

mes	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
N° de Clientes adicionales	1	2	2	3	2	3	3	4	4	5	4	5
Valor por Clientes	1000000	2000000	2000000	3000000	2000000	3000000	3000000	4000000	4000000	5000000	4000000	5000000

Figura 4: Flujo de caja sin proyecto.

Anexo 4: Flujo de caja con Proyecto.

	Inversión	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
N° de Clientes adicionales		1	2	2	3	4	4	4	4	5	6	6	7
Valor por Clientes		1000000	2000000	2000000	3000000	4000000	4000000	4000000	4000000	5000000	6000000	6000000	7000000
Costo FireBase	-864000	-18000	-36000	-36000	-54000	-72000	-72000	-72000	-72000	-90000	-108000	-108000	-126000
Valor generado por el proyecto	-864000	0	0	0	0	2000000	1000000	1000000	0	1000000	1000000	2000000	2000000
Tasa		12% VAN			\$2.679.214,02 TIR		38%						

Figura 5: Flujo de caja con proyecto.

Anexo 5: Flujo de caja con Proyecto y estimación optimista.

	Inversión	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
N° de Clientes adicionales		1	2	3	4	4	5	6	7	7	8	9	10
Valor por Clientes		1000000	2000000	3000000	4000000	4000000	5000000	6000000	7000000	7000000	8000000	9000000	10000000
Costo FireBase	-1188000	-18000	-36000	-54000	-72000	-72000	-90000	-108000	-126000	-126000	-144000	-162000	-180000
Valor generado por el proyecto	-1188000	0	0	1000000	1000000	2000000	2000000	3000000	3000000	3000000	3000000	5000000	5000000
Tasa		12% VAN			\$26.812.000,00 TIR		55%						

Figura 6: Flujo de caja con proyecto y estimación optimista

Anexo 6: Flujo de caja con Proyecto y estimación pesimista.

	Inversión	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
N° de Clientes adicionales		1	2	2	3	2	3	3	4	4	5	4	5
Valor por Clientes		1000000	2000000	2000000	3000000	2000000	3000000	3000000	4000000	4000000	5000000	4000000	5000000
Costo FireBase	-684000	-18000	-36000	-36000	-54000	-36000	-54000	-54000	-72000	-72000	-90000	-72000	-90000
Valor generado por el proyecto	-684000	0	0	0	0	0	0	0	0	0	0	0	0
Tasa		12% VAN			\$-684.000,00 TIR		-						

Figura 7: Flujo de caja con proyecto y estimación optimista