

# ---Meerkats---

## Entregable 1: Definición del Problema

### Planteamiento del Problema

La plataforma de Italika no ofrece una experiencia y presentación personalizada de compra, y como consecuencia no se incentiva al cliente a concretar el proceso compra. Además, no se cuenta con un mecanismo capaz de analizar el nivel de satisfacción del cliente respecto a sus productos, por lo que no existe una base sólida para promocionar dichos productos a clientes potenciales. Estos factores ocasionan que el impacto de la marca no sea suficiente para consolidar un vínculo de lealtad con el cliente.

### Solución propuesta

Para solucionar la problemática descrita se propone una nueva experiencia de compra en dos canales principales, los cuales serán un sitio web y un bot de messenger. La característica de esta nueva experiencia de compra será la personalización de parámetros como el lenguaje usado, ordenamiento y sugerencias de productos. Además, con estos perfiles se valorará la posibilidad de ofrecer productos en crédito. Una vez concretada una compra se buscará obtener retroalimentación vía el chatbot de Messenger para obtener información de satisfacción del cliente y refinar las características de los perfiles de usuarios.

### Creación de perfiles de usuarios

Basados en la información del usuario y los datos históricos de compras se crearán distintos tipos de perfiles para que se pueda generar una atención personalizada basada en las categoría en la que entra el usuario.

### Experiencia de compra basada en el perfil de usuario

Con una interfaz que contiene los productos de Italika, crear un ambiente amigable y de atención personalizada para el usuario, el cual, mediante un bot basado en la información proporcionada por el usuario despliegue opciones de productos; métodos de pago como créditos y/o financiamientos para completar la venta.

# ---Meerkats---

## Chatbot en Messenger

Este módulo del sistema tendrá dos funciones: el guiar al usuario en el proceso de compra sin necesidad de salir de la aplicación de chat de Messenger y dar un seguimiento a la satisfacción del usuario respecto a los productos adquiridos. Esto, para ofrecer una experiencia más cercana de la marca con el cliente, preguntando si le ha gustado el producto, si ha tenido problemas, si ha ido a algún centro servicio y otras preguntas útiles para pulir los modelos que definen perfiles de usuarios. Con base en las respuestas de los usuarios hacer un análisis de sentimientos para mejorar la calidad de los productos y servicios ofrecidos.

## Tecnologías a utilizar

### Servicios

- Dialogflow
- Google App Engine
- PostgreSQL

### Bibliotecas

- ReactJS
- Django
- ScikitLearn
- Keras

### Lenguajes de programación

- Python
- Javascript

## ---Meerkats---

### Entregable 2: Acotar la solución

#### Alcance del Proyecto a Corto Plazo (Hackathon)

- Creación perfiles de usuario basados en edad, sexo, ocupación, ingresos, historial crediticio y otros valores para personalizar la atención al cliente.
- Crear un sitio web con atención personalizada utilizando un árbol de decisiones y un bot que se adapte al perfil del cliente y ofrezca los productos de interés.
- Ofrecer una API para integración del proceso de compra con distintos canales para mejorar la experiencia del usuario.
- Generar un bot de seguimiento en messenger para retroalimentar la satisfacción del cliente.

#### Alcance del Proyecto a Largo Plazo

- Personalizar toda la experiencia de venta y postventa en línea para brindar el mejor servicio para el cliente a fin de crear lealtad con la marca.
- La aplicación atenderá a los clientes con un voice chat implementando la tecnología de AWS Polly.
- Integración vía call center creada con twilio y AWS.

¿Tomas en cuenta las necesidades/realidad de los usuarios del mismo? ¿Cómo?

Si, al acceder al sitio web se le preguntará al usuario: ¿Quieres recibir ayuda personalizada?. En caso de que la respuesta sea sí. Nuestro bot hará una serie de preguntas al usuario como: edad, ocupación, sexo, uso que le dará a la motocicleta. Mediante el uso, análisis y filtración de los datos anteriores, junto con la retroalimentación obtenida por parte de nuestros clientes, Italika tendrá la capacidad de segmentar su mercado meta, permitiendo la creación u oferta de productos que sean más atractivos y se adecuen a las necesidades de los

## ---Meerkats---

clientes. Buscando así maximizar el costo-beneficio y mejorando la experiencia y satisfacción del usuario.

Por otro lado, los métodos de pago actuales (Tarjetas Bancarias, Efectivo) serán complementados con créditos y descuentos personalizados, los cuales dependerá exclusivamente de la lealtad del cliente. Estos descuentos serán generados gracias a la información recolectada por el chatbot y el refinamiento de los perfiles de usuarios con el historial de compras, ingresos, intereses y buró de crédito.

## Arquitectura Técnica del Proyecto

El sistema estará construido con los siguientes puntos:

- Con el patrón de arquitectura Modelo Vista Controlador MVC.
- El framework para aplicaciones Django, el cual estará alojado en una arquitectura integrada con los servicios de Amazon, AWS EC2 y AWS Elastic Beanstalk.
- Funciones lambda de Amazon para obtener las predicciones de los modelos de Machine Learning y los servicios de bot a través de una conexión con un webhook.
- Se creará una instancia en Amazon RDS, de AWS, para nuestro gestor de base de datos Postgresql.
- El manejo de datos estáticos se hará a través de buckets de s3 de amazon.

## ---Meerkats---

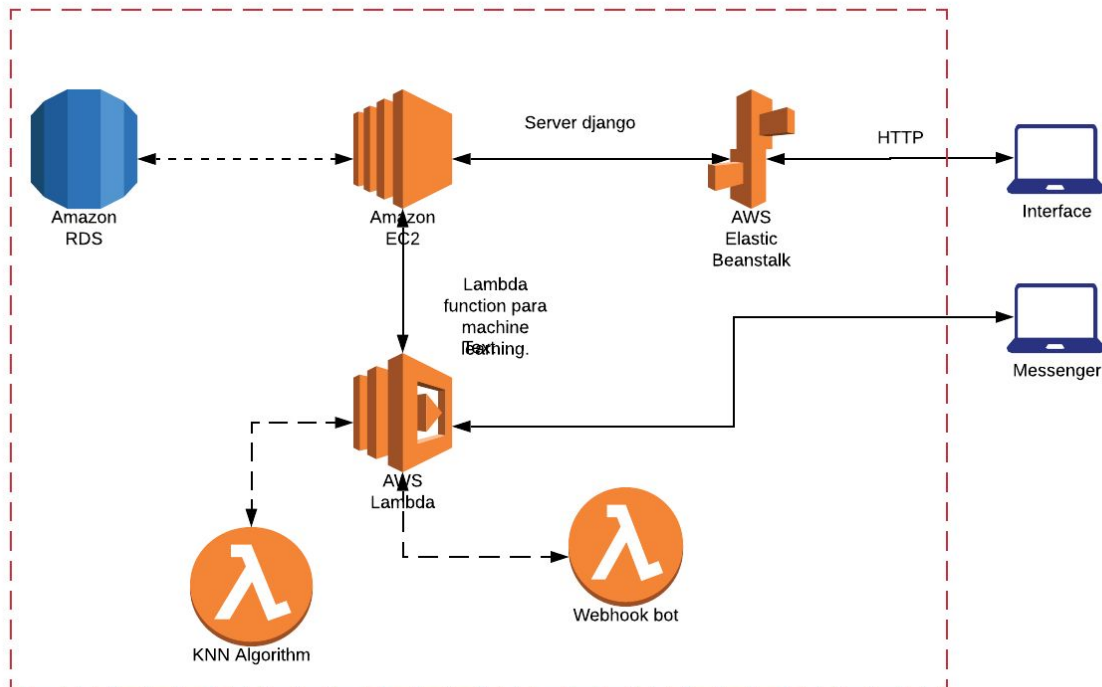


Figura 1.1. arquitectura prototipo del sistema.

## Entregable 3: Release

### ¿Qué aprendiste durante la primer edición del #GSHackChallenge?

En el transcurso del #GSHackChallenge aprendimos tanto del área técnica como de la creación de proyectos. En el área técnica aprendimos a utilizar Amazon Web Services tanto para poner los proyectos en producción, además de utilizar sus posibilidades de ejecutar código Python de manera eficiente y así poder integrar nuestros modelos de Machine Learning. Además, aprendimos a implementar un ChatBot que aunque en principio es una tarea relativamente simple, la integración del mismo en distintas plataformas resultó la parte más complicada. En lo que respecta a conocimiento sobre creación de proyectos aprendimos que desde un principio tienen que ser definidos de manera clara y concreta el problema, objetivos y MVP, para que a lo largo del desarrollo sean claros los objetivos a alcanzar.

## ---Meerkats---

Liga repositorio en github con acceso público para evaluación de código.

[https://github.com/Huitzoo/salinas\\_challenge](https://github.com/Huitzoo/salinas_challenge)

¿Cómo se organizaron para ejecutar el proyecto?  
¿Cómo dividieron las responsabilidades?

Dividimos la ejecución del proyecto en tres procesos: definición, documentación y desarrollo.

En la definición del proyecto la responsabilidad de ir a las pláticas informativas y conocer a fondo los retos planteados fue tomada por nuestro líder de equipo Oscar Chávez. Una vez conociendo los retos a fondo decidimos hacer una lluvia de ideas, donde participamos todos los miembros y el encargado de tomar nota de lo acordado en esta conversación del rumbo del proyecto fue Oscar Castillo. Después de haber sido planteadas las ideas del proyecto, se buscó mentoría de parte de Italika para refinar prioridades y así poder pasar a la fase de documentación.

En el proceso de documentación se siguió un flujo de trabajo donde habiendo decidido los productos a entregar y el problema a resolver, se dejó decidir a cada miembro del equipo las secciones de su interés, en caso de que más de un miembro tuviera interés en una sección se realizaba en conjunto. Una vez concluida una sección los miembros del equipo daban retroalimentación sobre la redacción y contenido de la misma, pudiendo así mejorar el contenido de los entregables y cuidando mantener el enfoque del proyecto en cada entregable.

El último proceso fue el de desarrollo, que se empezó de manera paralela con el entregable 2. En desarrollo decidimos dividirnos en dos equipos, Aiko y Daniel diseñaron el frontend y el equipo encargado del backend fue conformado por Jorge, Oscar Castillo y Oscar Chávez.

Dentro del equipo de backend. Las funciones principales dentro del equipo de backend se dividieron de la siguiente manera: Jorge y Oscar Castillo se encargaron de definir los datos a utilizar para entrenar al modelo para generar

## ---Meerkats---

perfiles de usuarios. Oscar Chávez sugirió utilizar un KNN y de ahí dividimos el backend teniendo trabajo individual. Oscar Castillo se encargó de extender el dataset obtenido para poder entrenar y generar el modelo KNN, este modelo genera las sugerencias de modelos de motocicletas según los datos del usuario. Jorge se encargó de definir los chatbots en Dialogflow. Oscar Chávez se encargó de levantar la arquitectura y la estructura del proyecto Django, además fue el encargado de la integración de los distintos servicios necesarios para utilizar el modelo KNN y los chatbots.

En la parte de frontend, Aiko López y Daniel Herrera se encargaron de la búsqueda e implementación de un template para una presentación adecuada de los datos y de la solución propuesta, en este se busco mejorar la experiencia con el usuario creando un entorno amigable e innovador. Previo a esto se encargaron de la búsqueda de datos reales y estadísticos, en páginas como la INEGI sobre nuestro clientes potenciales. En nuestro caso era de vital importancia conocer el sueldo de los clientes de acuerdo a su profesión. Los cuales fueron tratados y enviados a Oscar Castillo y Jorge para generar el modelo KNN.

¿Qué otras funcionalidades le añadirían al proyecto? Si tuvieran más tiempo

- La principal funcionalidad a integrar sería desarrollar una aplicación para Android y iOS, con posibilidades de ofrecer una experiencia basada en los requerimientos de Italika.
- Crear una UX con mayor tiempo nos permitiría diferenciar y personalizar aún más aspectos de la misma.
- Se integraría con el chatbot la capacidad de un voice chat implementando la tecnología de AWS Polly, para la inclusión de más sectores de la población.
- Integración de atención vía bot de call center creada con twilio y AWS, con esto se podría concretar una venta o recibir retroalimentación del usuario sin necesidad de datos móviles.

## ---Meerkats---

¿Está listo para producción? o ¿Qué hace falta para poder implementar?

El proyecto se encuentra listo para producción, actualmente se encuentra en un entorno de producción en AWS. Consideramos que lo necesario para tener un proyecto usable sería realizar tests unitarios y de integración, no realizados principalmente por cuestiones de tiempo. En lo que respecta a una integración con la plataforma actual de Italika, es necesario actualizar el modelo con los datos reales así como integrar el chatbot en la página de Facebook de Italika. En principio el modelo fue construido de manera general para ser fácilmente adaptable y en lo que respecta al chatbot, ya está en Webhooks por lo que resulta integrable con múltiples plataformas, como Facebook Messenger, Whatsapp, sitios web o medios con soporte de Webhooks.