

# Requerimientos Tecnológicos del Proyecto “Swivel”

Pulse <3 Studios

## Versionamiento

Versión	Fecha	Responsable	Comentarios
1.0.0	5/5/2023	Sebastián González	Creación del Documento
1.0.1	5/19/2023	Sebastián González	Formateo del Documento
1.1.0	5/19/2023	Sebastián González	Redacción de Secciones
1.2.0	6/7/2023	Sebastián González	Actualización de diagrama, adición de OpenAI

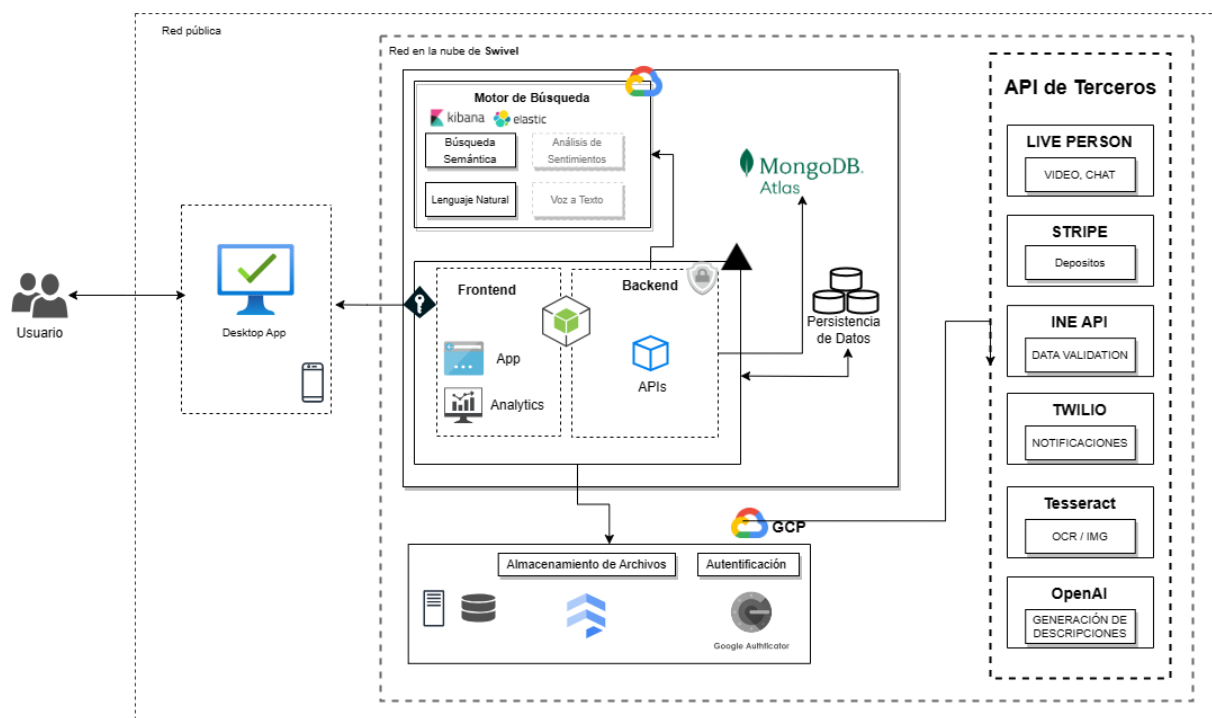
# Índice

<b>Arquitectura del Sistema</b>	<b>3</b>
Red Pública	3
Usuario	3
Red en Nube de Swivel	3
Frontend	4
Backend	4
Base de Datos	4
Autenticación	4
Almacenamiento de Archivos	4
Motor de Búsqueda	4
APIs de Terceros	4
Live Person	4
Stripe	5
Verificamex	5
Twilio	5
OpenAI	5

## AGREGAR INTRODUCCIÓN

# Arquitectura del Sistema

En esta sección se presenta un diagrama ilustrativo de la arquitectura que describe la aplicación y los diferentes servicios que se utilizan para su despliegue y desarrollo. Este incluye los elementos de Usuario, Frontend, Backend, APIs de Terceros y las conexiones entre estas. De igual manera se describe cada uno de los componentes a profundidad, haciendo énfasis en las tecnologías que se utilizan y su conexión con las demás partes.



*Fig. 1*  
Diagrama de Arquitectura

## Red Pública

### Usuario

Los usuarios acceden a nuestra aplicación e infraestructura a través de un navegador ya sea en su computadora personal o su teléfono móvil. La aplicación es compatible con la mayoría de los navegadores modernos gracias a las tecnologías usadas para su desarrollo.

## **Red en Nube de Swivel**

Dentro de este segmento se encuentra el frontend, backend, la base de datos, el motor de búsqueda y todos los demás servicios que componen la aplicación.

### **Frontend**

El “frontend” de la aplicación, es decir la interfaz de usuario ,consta de la aplicación de compra y venta de autos y la parte de analitica de datos. Toda esta parte está desarrollada utilizando NodeJS, específicamente el framework NEXT. Se utilizará Vercel para desplegar el servicio y hacerla accesible fuera de la red privada de Swivel. Además se considera un sistema de autenticación y persistencia de datos para el manejo de las sesiones y los permisos que tiene cada usuario.

### **Backend**

El backend de la aplicación está constituido por todos aquellos módulos que permiten extraer información de la base de datos y otros recursos y desplegarla en el frontend. Para su desarrollo se usó NodeJS y NEXT al igual que el frontend. Debido a que tanto frontend como backend forman parte de una misma aplicación, esta parte también estará desplegada en Vercel. Al igual que frontend considera la autenticación y persistencia de datos para las sesiones y usuarios.

### **Base de Datos**

La base de datos que se utiliza en el sistema es MongoDB, desplegada en un servicio propio de mongo llamado Mongo Atlas. Esta base de datos se conecta directamente con los APIs que constituyen al backend y guarda toda la información de la aplicación a excepción del catálogo.

### **Autenticación**

Para la autenticación se utilizan dos servicios. El primero es la librería para el framework en el que está desarrollada la aplicación, NextAuth. Este permite el manejo de los permisos y sesiones del usuario en toda la aplicación y permite almacenar las credenciales de estos en nuestra propia base de datos de manera segura. El segundo servicio es Google Authenticator, el cuál permite al usuario usar su cuenta existente de Google para crear una cuenta en el sistema.

### **Almacenamiento de Archivos**

Para el almacenamiento de archivos se utiliza Firestore, un servicio de almacenamiento que forma parte de la nube de Google. Este se conecta con los APIs del backend para procesar el archivo y finalmente guardar la liga del archivo en la base de datos.

### **Motor de Búsqueda**

Para el motor de búsqueda se hace uso de ElasticSearch. Este recurso permite habilitar las funcionalidades de búsqueda por filtros y por lenguaje natural y abre la posibilidad de utilizar otras como lo son voz a texto e integración de modelos pre-entrenados para mejorar la búsqueda. El servicio se encuentra desplegado en la nube de Google y es consumido por medio de las APIs del backend. Este motor de búsqueda almacena únicamente el catálogo de autos y la información relacionada a ellos, permitiendo su fácil recuperación y acceso.

## **APIs de Terceros**

### **Live Person**

La aplicación hace uso de Live Person, un servicio de comunicación en vivo que permite chat, llamada y videollamada en tiempo real. Este servicio se utiliza para el soporte de la aplicación.

### **Stripe**

Para la integración de pagos en la plataforma se utiliza Stripe. Este servicio permite generar una pantalla de “checkout” para que el usuario pueda hacer el pago del auto que está comprando. Este servicio se conecta con el frontend para posteriormente guardar las claves de rastreo del pago en la base de datos.

### **Verificamex**

Este servicio es utilizado para la verificación de documentos de identificación del usuario, específicamente la INE.

### **Twilio**

Twilio permite mandar notificaciones ya sea por mail, whatsapp o cualquier otro medio de contacto al usuario.

### **OpenAI**

El API de OpenAI permite utilizar sus modelos generativos de inteligencia artificial para generar las descripciones para cada auto en el catálogo. Estas descripciones son las que nutren la búsqueda por lenguaje natural en ElasticSearch.