Segunda Fase (CyberBasket Solutions)

**Juan Sebastian Henao Parra**

**Sebastian Galvez Yepes**

**Luis Fernando Gonzalez Zambrano**

**Juan Pablo Pulgarin Musicue**

**Jhony Alejandro Perez Gomez**

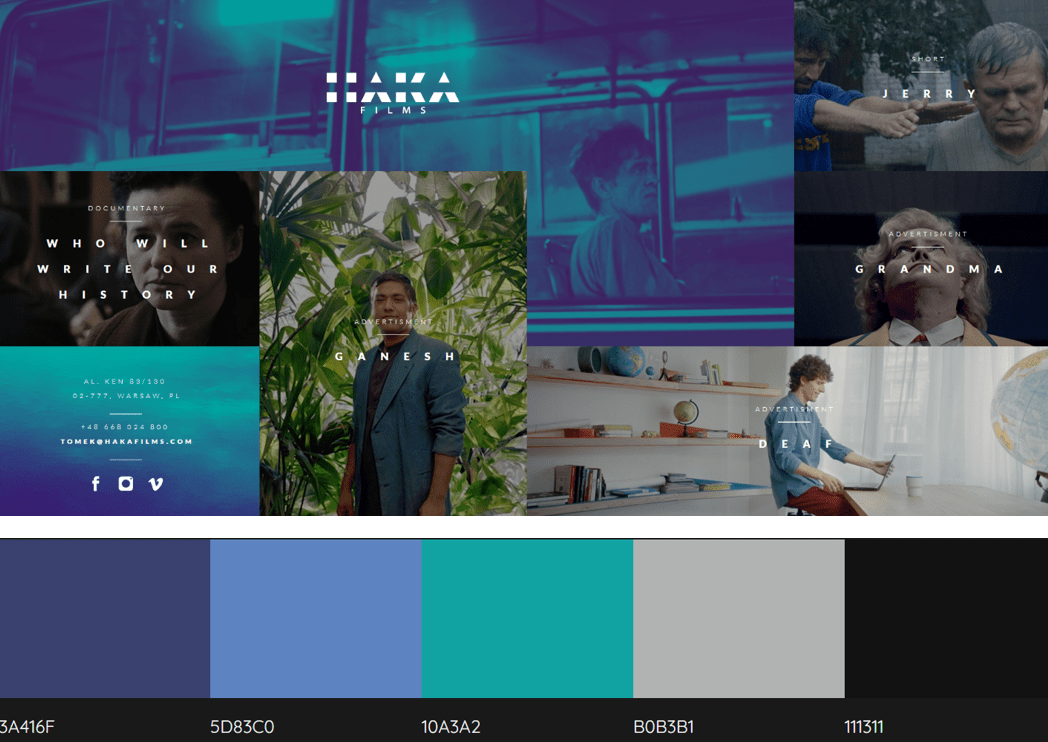
**Introducción**

La segunda fase del desarrollo de una aplicación es la fase de diseño. En esta fase, se definen los aspectos visuales y funcionales de la aplicación. Los mockups de diseño son una herramienta esencial en esta fase, ya que sirven prototipos de la aplicación que se realizan con herramientas de diseño gráfico, en nuestro caso se desarrollaron con Figma

Los mockups permiten visualizar cómo va a ser la aplicación antes de que se comience a programar. Esto es importante porque permite obtener retroalimentación de los usuarios y los desarrolladores antes de que se invierta mucho tiempo y dinero en la programación.

La retroalimentación sobre el diseño de la aplicación puede ser útil para identificar problemas potenciales, mejorar la usabilidad y la experiencia del usuario, y garantizar que la aplicación cumple con los requisitos de los usuarios.

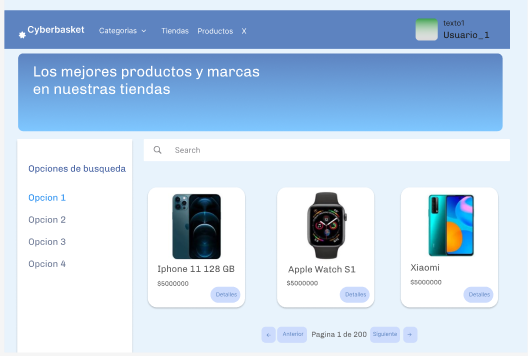
Para este diseño se definió la siguiente paleta de colores.



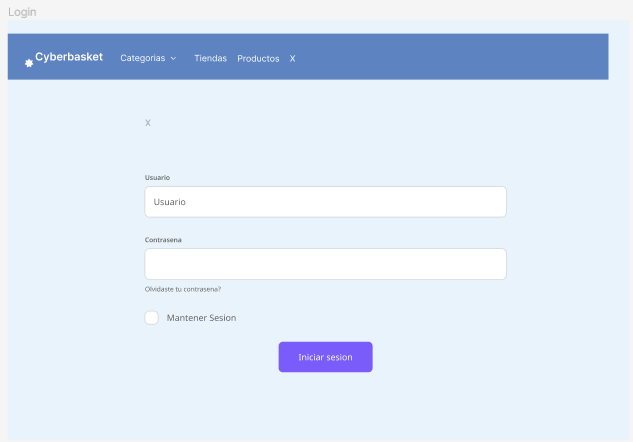
Para los mockups se definieron los siguientes mockups  
  
**Home:**

Este caso de uso describe el comportamiento de la pantalla de inicio de la plataforma, que muestra a los usuarios una vista general de la página web y les permite acceder a sus funciones principales.  


**Búsqueda:**

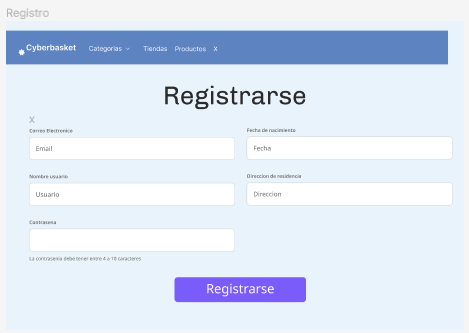
Los usuarios pueden buscar productos en la plataforma CyberBasket.  


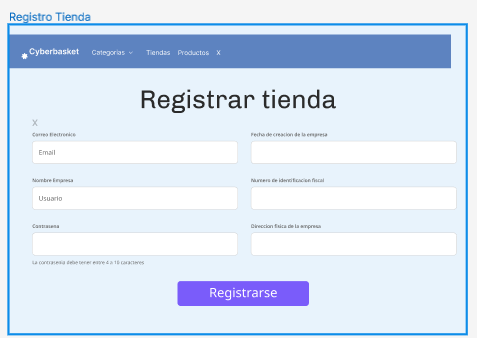
**Login:**

Este caso de uso describe el proceso de inicio de sesión para los usuarios que ya se han registrado en la plataforma, asegurando un acceso rápido y seguro a la página web.  


**Registro:**

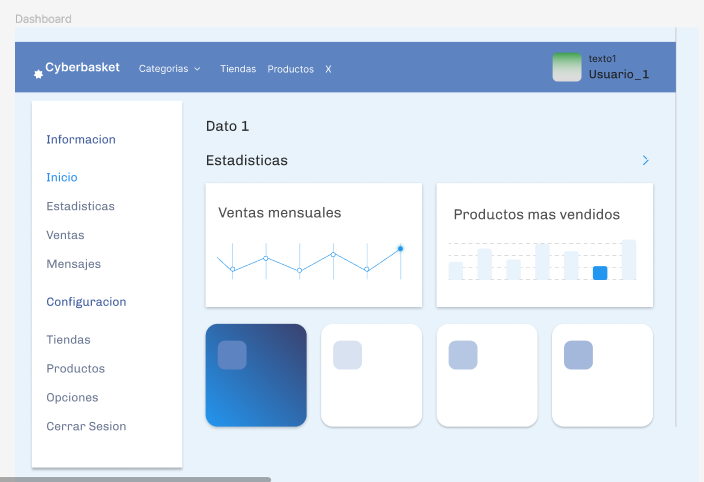
Los usuarios pueden registrarse en la plataforma CyberBasket para crear una cuenta.



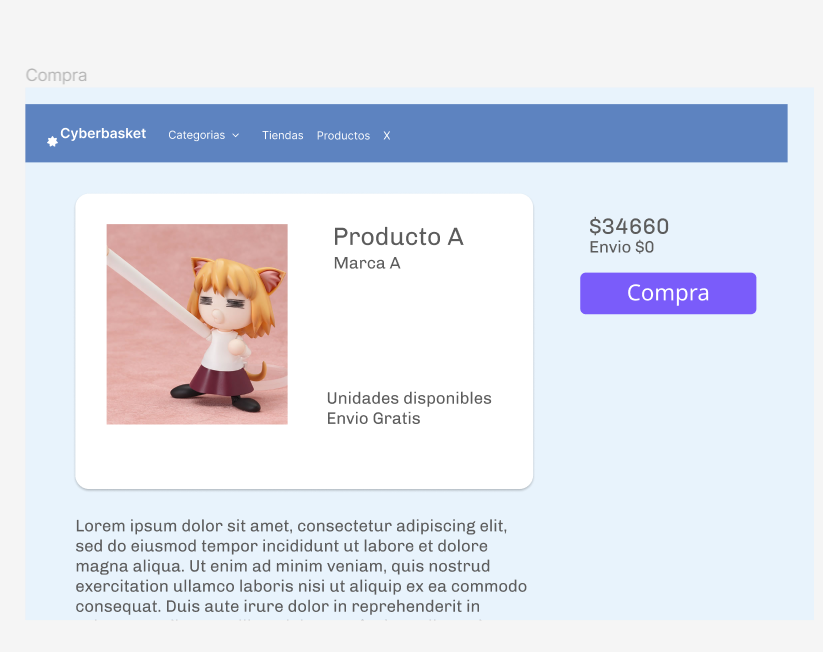


**Dashboard:**

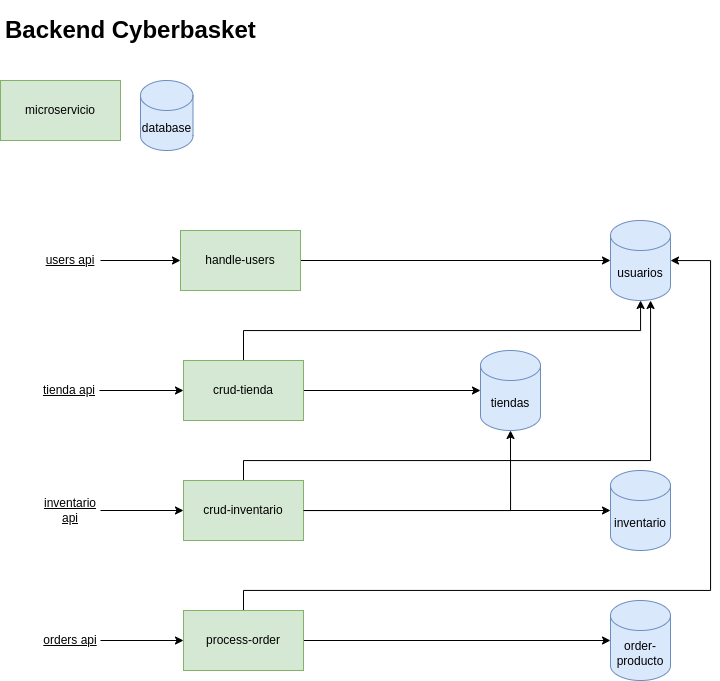
Este caso de uso describe el proceso de visualización de datos en una pantalla de dashboard, proporcionando información relevante para los usuarios según su rol, por ahora se muestra el de Vendedores.



**Carrito de compras y venta:**

Este caso de uso describe el proceso de realizar una compra en la plataforma CyberBasket, desde la selección de productos hasta el pago y confirmación de la compra.

**Diagrama de arquitectura:**

La arquitectura MVC es un patrón de diseño para aplicaciones web que separa la lógica de presentación de la lógica de negocio. En este caso, la página web será la interfaz de usuario que se encarga de mostrar la información al usuario y de recoger sus datos. Los microservicios serían los encargados de realizar las operaciones de negocio, como almacenar datos, procesar transacciones o realizar llamadas a APIs externas.  
  


**Base de datos:**

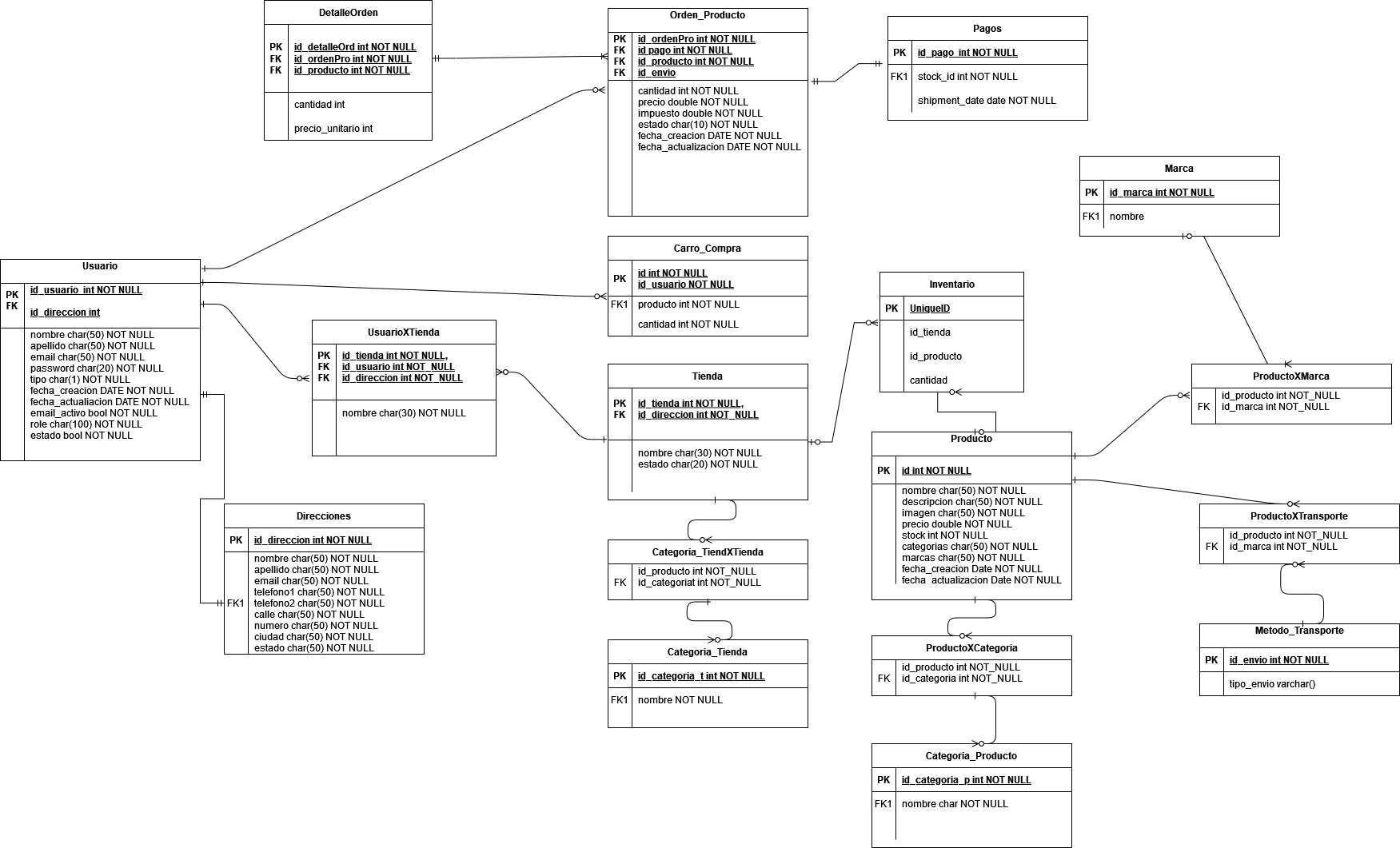
En la base de datos, se pueden visualizar las conexiones entre los componentes para poder tener una estructura en la página. Por ejemplo, si la página tiene una sección para usuarios, una sección para productos y una sección para órdenes. Estas tablas estarán relacionadas entre sí, por ejemplo, una tabla de órdenes puede tener una columna que referencia la ID de un usuario.

La idea general es usar FastApi como framework del backend para las conexiones y comunicación con la base de datos. FastApi es un framework de Python que facilita el desarrollo de APIs RESTful. Las APIs RESTful son un tipo de API que utiliza el protocolo HTTP para comunicarse.

En este caso, FastApi se puede utilizar para crear una API que permita acceder a los datos de la base de datos. La API podría tener rutas para obtener, crear, actualizar y eliminar datos.

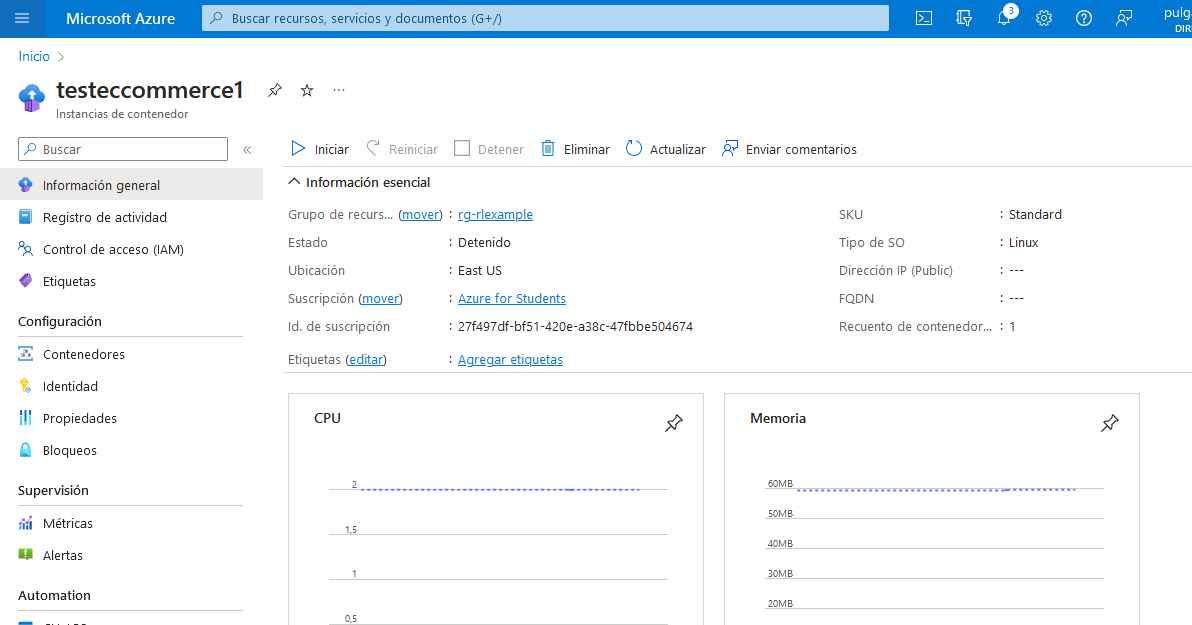
Por ejemplo, la API podría tener una ruta /users que devuelva una lista de usuarios. La API también podría tener una ruta /users/create que permita crear un nuevo usuario.

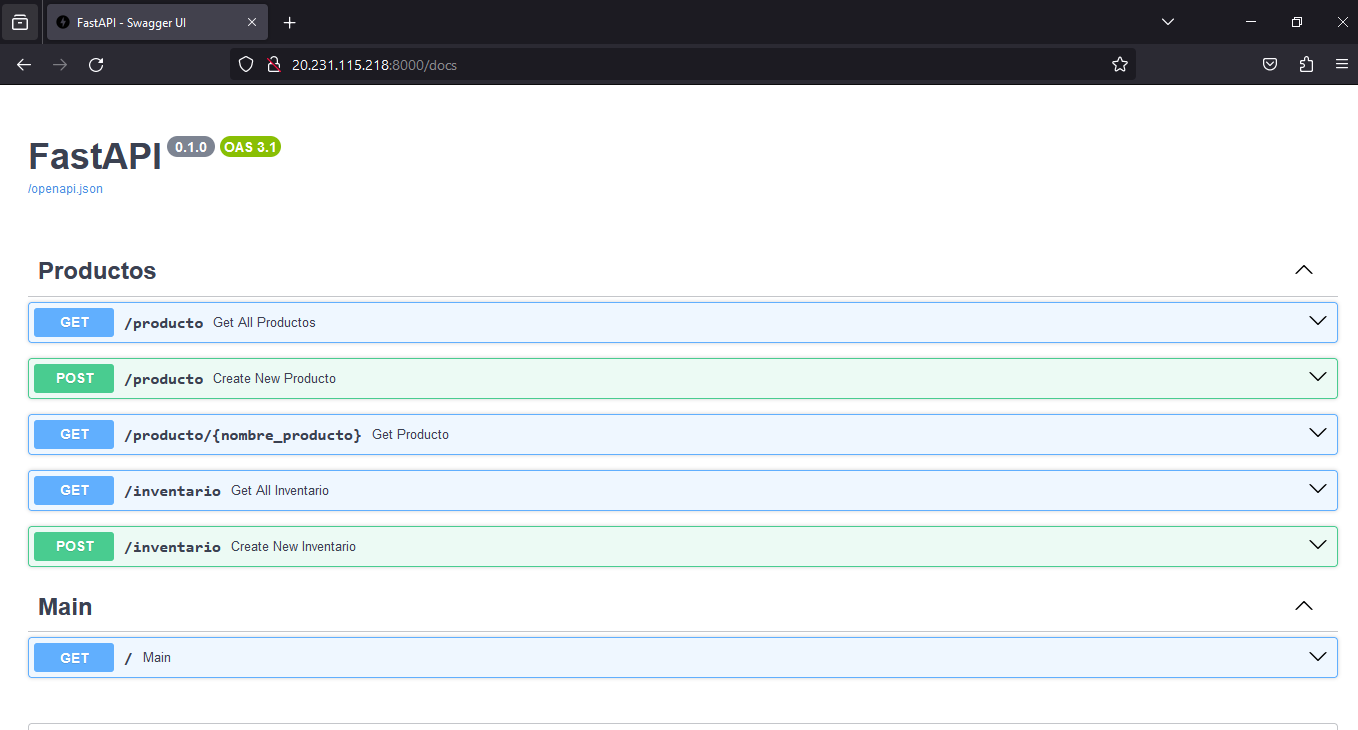
El frontend de la aplicación puede utilizar la API para acceder a los datos de la base de datos. El frontend podría utilizar los datos para mostrar la estructura de la página.



**Demo de despliegue de Microservicio:**

Para probar la funcionalidad de la base de datos se usó un prototipo de backend, con dos tablas populadas subidas a un Docker en Azure, este sistema se hizo como prueba para ver si una plataforma como Azure era útil para nuestras necesidades





**Conclusión de la fase 2:**  
Luego de analizar los conceptos, se concluyó que por la estructura y la base de datos, las plataformas que se usarán serán Angular (para el desarrollo del front) y se usarán microservicios que se subirán a la plataforma de Azure. Estos microservicios estarán diseñados para compartir la misma base de datos, lo que permitirá una mayor escalabilidad y flexibilidad del sistema.