Tercera Fase(CyberBasket Solutions)

**Juan Sebastian Henao Parra**

**Sebastian Galvez Yepes**

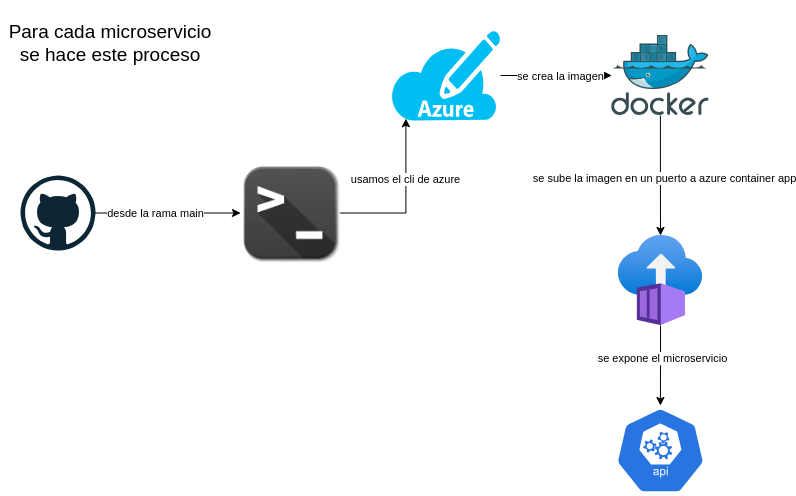
**Luis Fernando Gonzalez Zambrano**

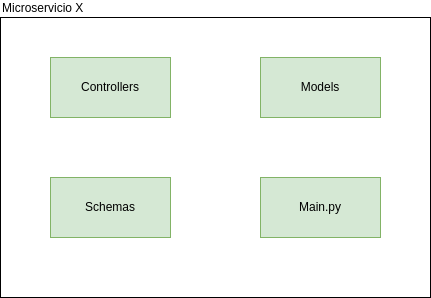
**Juan Pablo Pulgarin Musicue**

**Jhony Alejandro Perez Gomez**

https://github.com/JuanIIDX/CyberBasket

Para la plataforma de comercio electrónico CyberBasket se ha planteado la arquitectura de microservicios para proporcionar una mayor escalabilidad, flexibilidad y facilidad de mantenimiento. Los microservicios son aplicaciones independientes que se comunican entre sí a través de API.





La arquitectura de microservicios de CyberBasket se compone de los siguientes componentes:

**Frontend:** El frontend es la parte de la aplicación que interactúa con los usuarios. Está implementado en Angular y se ejecuta mediante Angular Web Services.

**Backend:** El backend es la parte de la aplicación que realiza el procesamiento de datos. Está implementado en FastAPI y se ejecuta en Azure SQL Database.

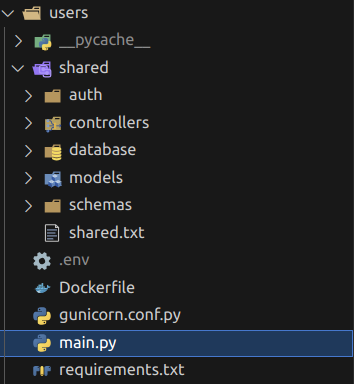
**Base de datos:** La base de datos almacena los datos de la aplicación, se utiliza Azure SQL Database.

La razon por la cual se escogió la arquitectura de microservicios de CyberBasket es ya que este ofrece una serie de beneficios como:

**Escalabilidad:** Los microservicios se pueden escalar de forma independiente, lo que permite a la plataforma adaptarse a las necesidades cambiantes del negocio.

**Flexibilidad:** Los microservicios se pueden desarrollar y desplegar de forma independiente, lo que facilita la adición de nuevas funcionalidades y la corrección de errores.

**Facilidad de mantenimiento:** Los microservicios son más fáciles de mantener que las aplicaciones monolíticas, ya que cada microservicio es responsable de una sola funcionalidad.



Los microservicios del Backend se organizan en torno a las siguientes funcionalidades:

**Inventario:** Este microservicio se encarga de gestionar los productos de la pagina, su busqueda e información.

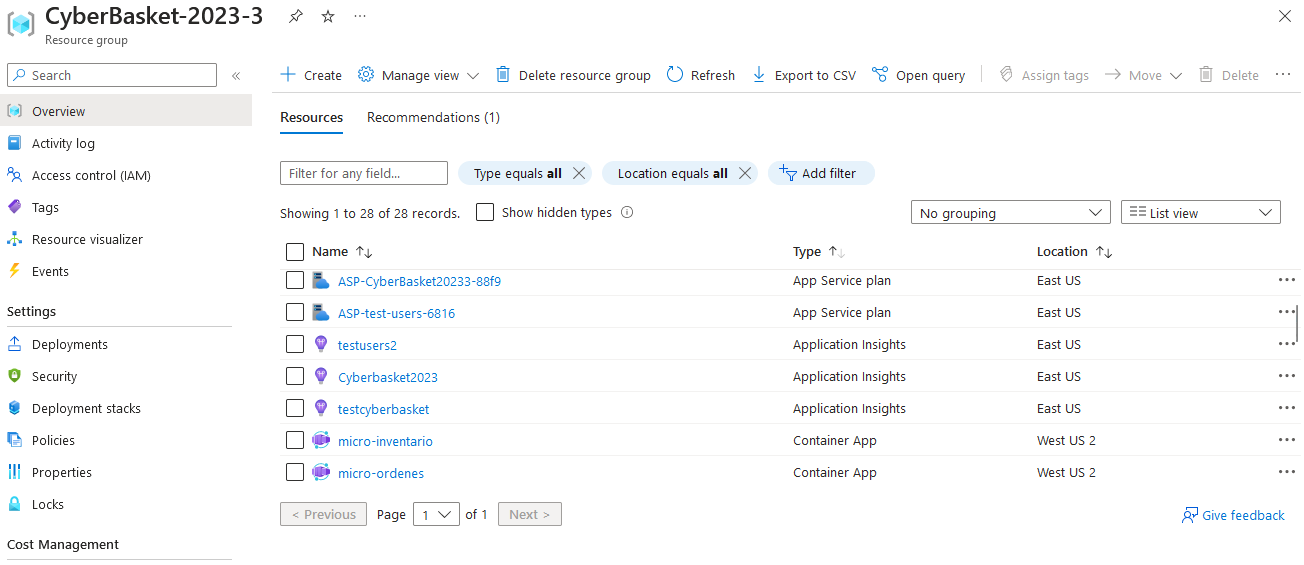
**Tienda:** Este microservicio se encarga de las tareas de la tienda, muestra informacion y estadisticas.

**Procesamiento de pagos:** Este microservicio se encarga de procesar los pagos realizados por los clientes.

**Ordenes:** Este microservicio se encarga del procesamiento de informacion de las ordenes

**Usuario:** Este microservicio se encarga del ingreso de los usuarios, devuelve un token.

Para el caso de CyberBasket, se utilizó el portal de Azure para crear el grupo de recursos principal. El grupo de recursos es un contenedor que agrupa los recursos de Azure relacionados.



En nuestro caso, el grupo de recursos principal contiene los siguientes recursos:

**Un servidor web:** El servidor web proporcionará la infraestructura necesaria para alojar la aplicación web de CyberBasket.

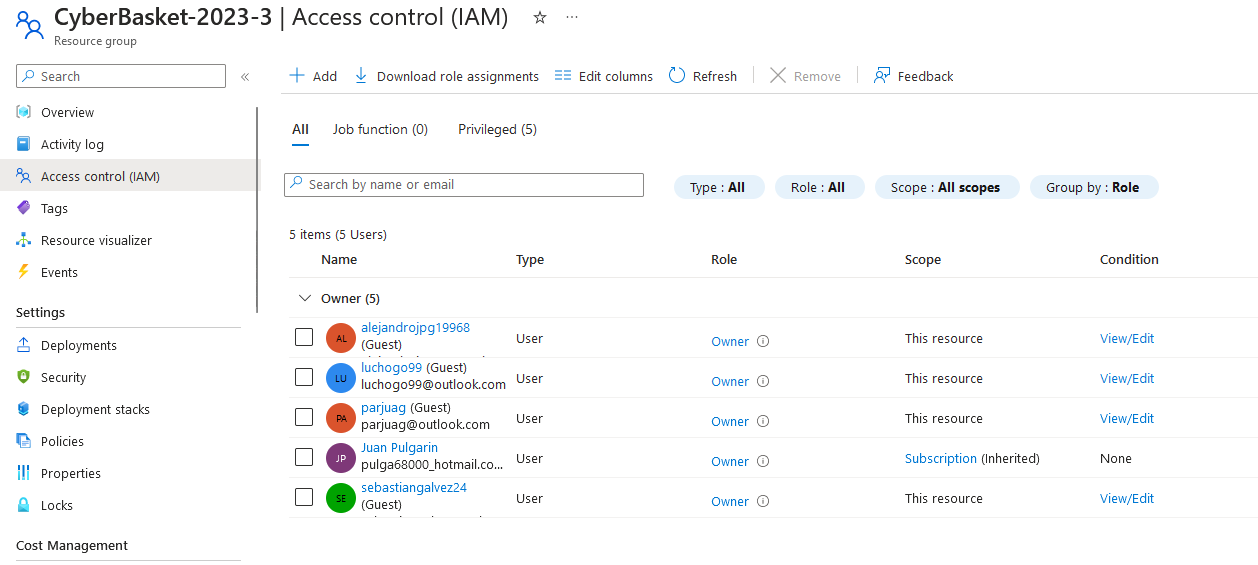
**Una base de datos:** La base de datos almacenará los datos de la aplicación web, como los productos, los clientes y las transacciones.

**Contenedores para los microservicios:** Contenedores de Docker el cual cada uno tiene un microservicio

Para la creación de esto, pensamos en los siguientes puntos

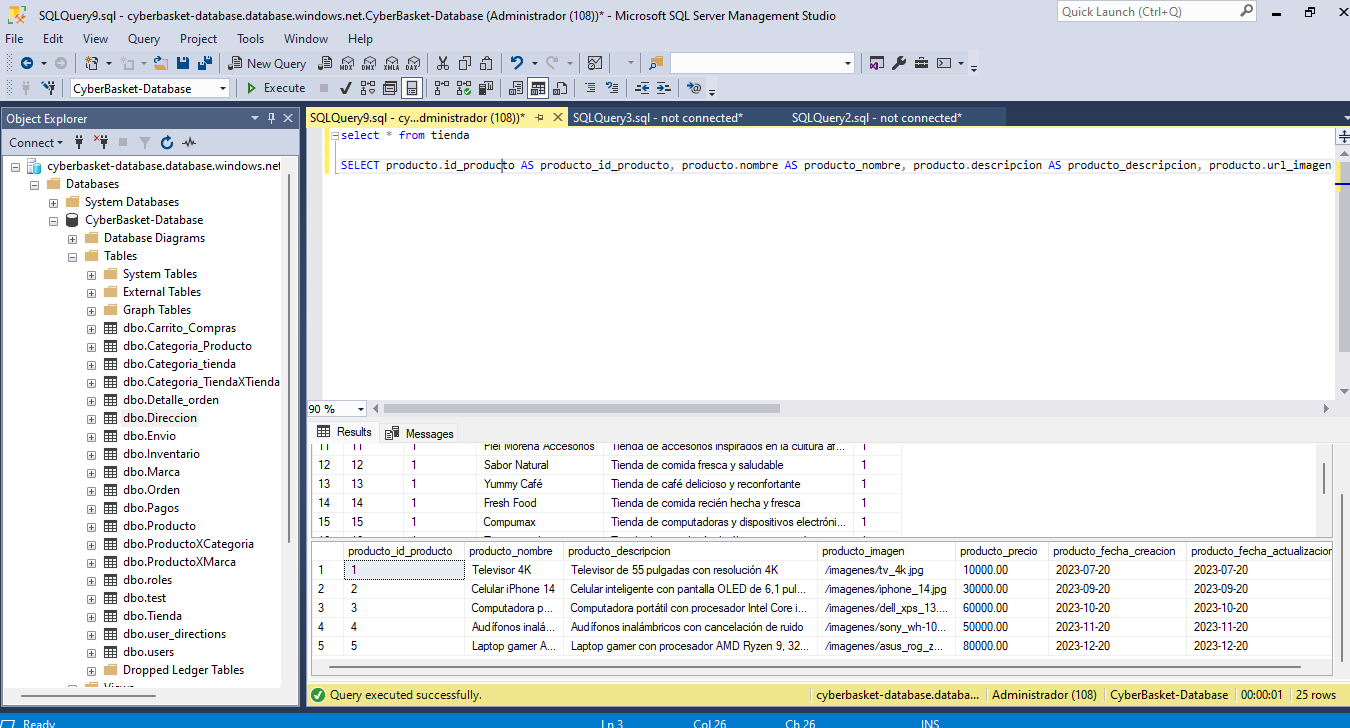
* El servidor web debe ser escalable para poder tener nuevas funcionalidades.
* La base de datos debe ser segura para proteger los datos de los clientes.
* Los microservicios deben ser estables y no tener problemas de conexión

Para organizar los permisos, el equipo utilizó un sistema de control de acceso basado en roles. Ya que cada uno debía tener acceso total a todos los recursos del contenedor, se asignó el mismo rol a todos los del grupo, durante la creación de algunos recursos existieron errores de permisos y problemas técnicos, las cuales requiriendo la intervención de todo el grupo para ser solucionadas.



**Como se creo la base de datos**

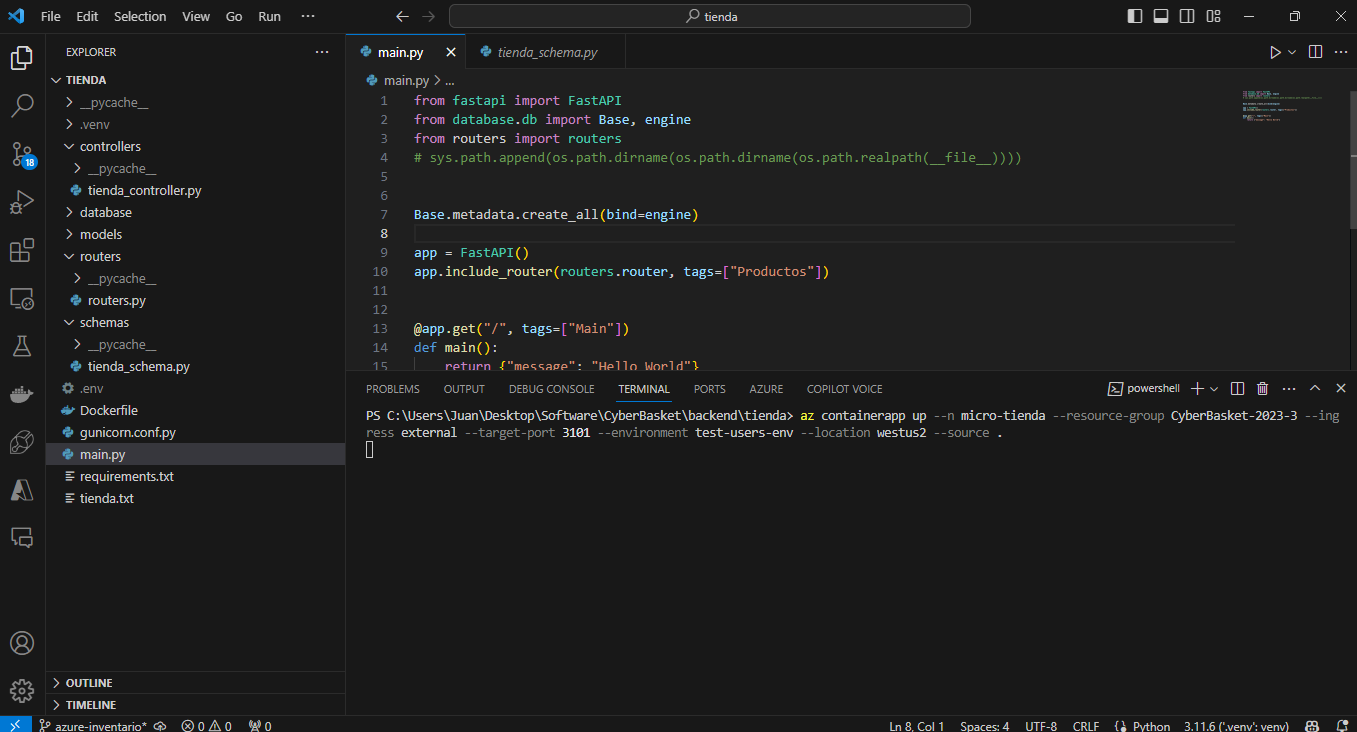
Se revisaron los requisitos para identificar los datos que necesitaba almacenar la base de datos. Estos datos incluían información sobre clientes, productos, ordenes entre otros, a partir de aquí se crearon las tablas, una vez que se identificaron los datos necesarios, se procedió a diseñar la estructura de la base de datos. Esto se hizo utilizando un diagrama de entidad-relación (ERD)

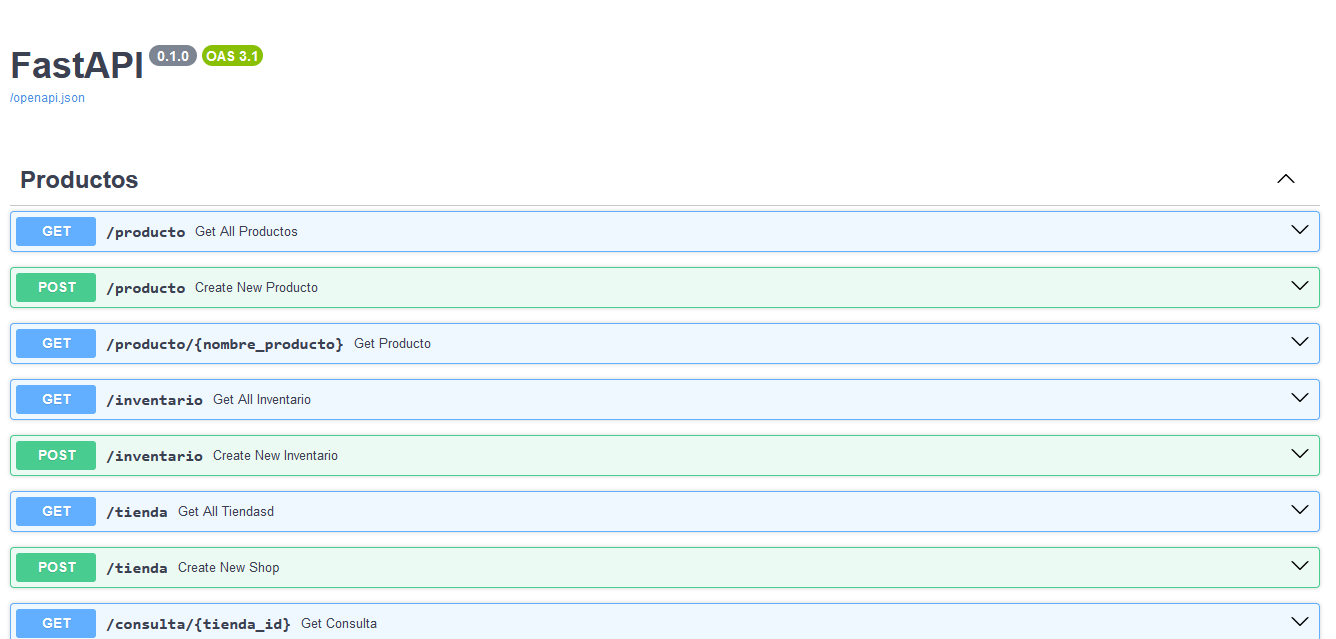


Una vez que se diseñó la estructura de la base de datos, se procedió a implementar por medio de Microsoft SQL server debido a su facil implementacion y corrección facil de errores. Luego de implementarse la base de datos, se realizaron pruebas para garantizar que funcionara correctamente. Las pruebas incluyeron pruebas unitarias, pruebas de integración y pruebas de aceptación por medio de consultas.

**Se crearon los microservicios esenciales para la aplicación y se desplegaron en azure**

Luego de esto, se organizaron los microservicios en fast Api, luego de comprobar su conexión a la base de datos, se desplegaron a Azure a través del servicio de contenedores, se usó el guvicorn para servir el fast api en un puerto específico(3100), luego se hizo el docker file y se expuso el puerto al que hace referencia el guvicorn, eso para cada microservicio.





Esto generaba links los cuales seran consultados a futuro por el Backend, cada microservicio creo un container app con el dockerfile y se subio usando el azure CLI, el despliegue se hace manual por ahora para hacer prueba, a futuro se implementara el github functions.

<https://micro-inventario.victoriouspebble-f396dfa4.westus2.azurecontainerapps.io/docs>

<https://test-users.victoriouspebble-f396dfa4.westus2.azurecontainerapps.io/docs>

<https://micro-tienda.victoriouspebble-f396dfa4.westus2.azurecontainerapps.io/docs>