La arquitectura de la solución incluye los siguientes componentes:

Blockchain:

- Backend: Desarrollado en Python, utiliza una base de datos SQL Server.
- Frontend: Desarrollado en Angular.
- El blockchain contiene las funcionalidades de inicio de sesión, creación de usuario, consulta de saldo y transacciones.

Snort y Snorby:

- Snort: Sistema de detección de intrusiones (IDS) basado en reglas, utilizado para el análisis y la detección de ciberataques.
- Snorby: Interfaz web utilizada para el análisis de tráfico y protocolos de red, y para la visualización de alertas y notificaciones generadas por Snort.

Docker:

- Se utiliza Docker para facilitar la implementación y el despliegue de los diferentes componentes de la solución.
- Los servicios relacionados con el blockchain se implementan en contenedores Docker separados:
    - **midb**: Contenedor para la base de datos SQL Server utilizada por el blockchain.
    - **miback**: Contenedor para el backend del blockchain desarrollado en Python.
    - **mifront**: Contenedor para el frontend del blockchain desarrollado en Angular.
- Los servicios relacionados con Snort y Snorby también se implementan en contenedores Docker separados:
    - **db**: Contenedor para la base de datos MySQL utilizada por Snorby.
    - **adminer**: Contenedor con una interfaz web para administrar la base de datos de Snorby.
    - **snorby**: Contenedor que ejecuta la aplicación Snorby, utilizada para el análisis de tráfico y protocolos de red.
    - **ids**: Contenedor que ejecuta Snort IDS para la detección y prevención de ciberataques.

Redes:

- Se establece una red de Docker llamada red_block para conectar los contenedores del blockchain.
- Los contenedores de Snort y Snorby utilizan la red predeterminada de Docker.

En resumen, la arquitectura incluye los componentes de un blockchain desarrollado en Python con backend y frontend separados, una base de datos SQL Server, Snort IDS para la detección de

ciberataques y Snorby para el análisis de tráfico y protocolos de red. Todos estos componentes se implementan y se ejecutan en contenedores Docker separados para facilitar la implementación y la gestión de la solución.