**Introducción**

El sistema de votación en Docker es una aplicación distribuida que permite a los usuarios votar por una de las dos opciones, 'a' o 'b'. La arquitectura se basa en contenedores Docker que se ejecutan en un entorno de orquestación de contenedores para garantizar la alta disponibilidad y escalabilidad.

**Componentes del Sistema**

El sistema consta de varios componentes que trabajan juntos para proporcionar una plataforma de votación eficiente y confiable:

1. **Servicio de Votación (Vote)**
   * Imagen Docker: Se basa en el código fuente del directorio **./vote** y utiliza Python para ejecutar la aplicación.
   * Puerto Externo: 5000
   * Funcionalidad: Proporciona la interfaz web para que los usuarios emitan votos.
   * Dependencias: Requiere que el servicio de Redis (**redis**) esté en funcionamiento.
2. **Servicio de Resultados (Result)**
   * Imagen Docker: Basado en el código fuente del directorio **./result**, utiliza Node.js para ejecutar la aplicación.
   * Puerto Externo: 5001
   * Funcionalidad: Muestra los resultados de las votaciones en tiempo real.
   * Dependencias: Requiere que el servicio de base de datos PostgreSQL (**db**) esté en funcionamiento.
3. **Servicio de Procesamiento (Worker)**
   * Imagen Docker: Se basa en el contexto del directorio **./worker** y se encarga del procesamiento en segundo plano.
   * Funcionalidad: Procesa y almacena los votos en la base de datos PostgreSQL.
   * Dependencias: Requiere que el servicio de Redis (**redis**) y el servicio de base de datos PostgreSQL (**db**) estén en funcionamiento.
4. **Servicio de Redis (Redis)**
   * Imagen Docker: Usa Redis en Alpine.
   * Funcionalidad: Almacena los votos temporalmente antes de ser procesados.
   * Dependencias: No depende de otros servicios.
5. **Servicio de Base de Datos PostgreSQL (DB)**
   * Imagen Docker: Utiliza PostgreSQL en Alpine.
   * Funcionalidad: Almacena los votos y resultados de votación.
   * Dependencias: No depende de otros servicios.
6. **Servicio de Semilla (Seed)**
   * Imagen Docker: Se encarga de poblar la base de datos con votos iniciales.
   * Funcionalidad: Ejecuta una vez para cargar datos de prueba en la base de datos.

**Interacción entre Componentes**

La arquitectura del sistema se caracteriza por la comunicación entre los componentes a través de solicitudes HTTP y el uso de una cola temporal (Redis) para procesar votos. Aquí se describe cómo interactúan los componentes:

1. Un usuario emite un voto a través de la interfaz web proporcionada por el servicio de Votación (Vote). El voto se envía al servicio de Redis para su almacenamiento temporal.
2. El servicio de Procesamiento (Worker) monitorea la cola de Redis y recupera votos pendientes. Luego, procesa los votos y los almacena en la base de datos PostgreSQL.
3. El servicio de Resultados (Result) consulta la base de datos PostgreSQL para obtener los resultados de votación y los muestra a los usuarios en tiempo real a través de una interfaz web.
4. El servicio de Semilla (Seed) se ejecuta solo cuando se requiere cargar datos de prueba en la base de datos.

**Conclusiones**

El sistema de votación en Docker utiliza contenedores para aislar y escalar sus componentes. Los servicios de Votación, Resultados, Procesamiento, Redis y la Base de Datos trabajan en conjunto para proporcionar una plataforma de votación en tiempo real. La arquitectura modular y bien orquestada garantiza que la aplicación sea confiable y escalable, lo que la hace adecuada para una variedad de aplicaciones de votación en línea.