



UNIVERSIDAD CENTRAL DEL ECUADOR
FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS
INGENIERÍA INFORMÁTICA
PROGRAMACIÓN DISTRIBUIDA

JUAN DIEGO GÓEZ MARTÍNEZ
JASON RIVERA PALMA

SISTEMAS DISTRIBUIDOS

Objetivos:

General:

El objetivo principal es investigar y comprender las características fundamentales de un sistema distribuido.

Específicos:

- Investigar las características clave de los sistemas distribuidos, como la descentralización, la comunicación entre componentes y la tolerancia a fallos.
- Estudiar cómo se dividen y organizan los componentes en un sistema distribuido y cómo interactúan entre sí.
- Identificar los beneficios clave de un sistema distribuido, como la escalabilidad horizontal, la alta disponibilidad y la redundancia.

Introducción

El sistema desarrollado es una aplicación de votación que permite a los usuarios emitir sus votos en una variedad de opciones. La aplicación está diseñada para funcionar de manera distribuida, lo que significa que se compone de múltiples componentes interconectados que colaboran para brindar la funcionalidad completa de la aplicación. Esto permite una mayor escalabilidad, redundancia y flexibilidad.

Tecnologías usadas:

- **Node.js:** Un entorno de ejecución de JavaScript que permite ejecutar código del lado del servidor. Node.js es ampliamente utilizado para la creación de aplicaciones web y servidores en tiempo real, como en tu caso para la implementación del servidor de votaciones.
- **Express.js:** Un marco de aplicaciones web para Node.js que facilita la creación de APIs y aplicaciones web con facilidad. Lo utilizaste para definir las rutas y la lógica del servidor en tu aplicación.

- **Socket.io:** Una biblioteca que permite la comunicación bidireccional en tiempo real entre clientes y servidores a través de WebSocket. Socket.io se usa para implementar la comunicación en tiempo real entre los clientes y el servidor en tu sistema de votaciones.
- **MongoDB:** Una base de datos NoSQL orientada a documentos. La utilizaste para almacenar los votos de los usuarios y mantener el estado de las votaciones.
- **Docker:** Una plataforma para desarrollar, enviar y ejecutar aplicaciones en contenedores. Usaste Docker para contenerizar tanto tu aplicación de servidor Node.js como la base de datos MongoDB, lo que facilita el despliegue en diferentes entornos.
- **Ionic:** Un framework para el desarrollo de aplicaciones móviles y web utilizando tecnologías web estándar como HTML, CSS y JavaScript. En tu caso, utilizaste Ionic para construir la interfaz de usuario de la aplicación móvil.
- **Angular:** Un framework de desarrollo de aplicaciones web y móviles basado en TypeScript. Ionic utiliza Angular para construir interfaces de usuario ricas y dinámicas en tu aplicación móvil.
- **Socket.io-client:** La biblioteca cliente de Socket.io para JavaScript, utilizada en la aplicación móvil para establecer conexiones de socket con el servidor y recibir actualizaciones en tiempo real.

Características de un sistema distribuido:

- **Descentralización:**
En el sistema de votaciones distribuido, la aplicación está compuesta por diferentes componentes: el servidor de votaciones, la base de datos y los clientes (aplicación Ionic). Cada uno de estos componentes puede operar de manera independiente y colaborar para brindar la funcionalidad completa. Por ejemplo, si un componente falla, como el servidor, otros componentes pueden seguir funcionando, asegurando que los usuarios aún puedan emitir votos.

- **Escalabilidad:**

La escalabilidad es fundamental en el sistema de votaciones, ya que el número de usuarios y la carga de trabajo pueden variar significativamente. El sistema distribuido permite agregar más instancias del servidor de votaciones y la base de datos según sea necesario para manejar picos de carga. Esto asegura que el sistema pueda seguir siendo eficiente y receptivo incluso cuando la demanda aumenta.

- **Tolerancia a Fallos:**

La tolerancia a fallos es esencial en un sistema de votaciones, ya que los usuarios deben poder emitir votos en todo momento. Si el servidor de votaciones o la base de datos experimentan un fallo, otros componentes pueden asumir sus funciones. Por ejemplo, si el servidor de votaciones falla, los clientes pueden conectarse a otro servidor disponible y continuar emitiendo votos sin interrupción.

- **Comunicación:**

En el sistema de votaciones distribuido, los componentes se comunican entre sí para lograr una experiencia fluida para los usuarios. Los sockets se utilizan para establecer conexiones bidireccionales en tiempo real entre el servidor y los clientes. Cuando un usuario emite un voto, se envía a través de un socket al servidor, que actualiza la base de datos y luego notifica a todos los clientes conectados sobre el nuevo voto. Esto asegura que todos los usuarios vean los resultados en tiempo real.

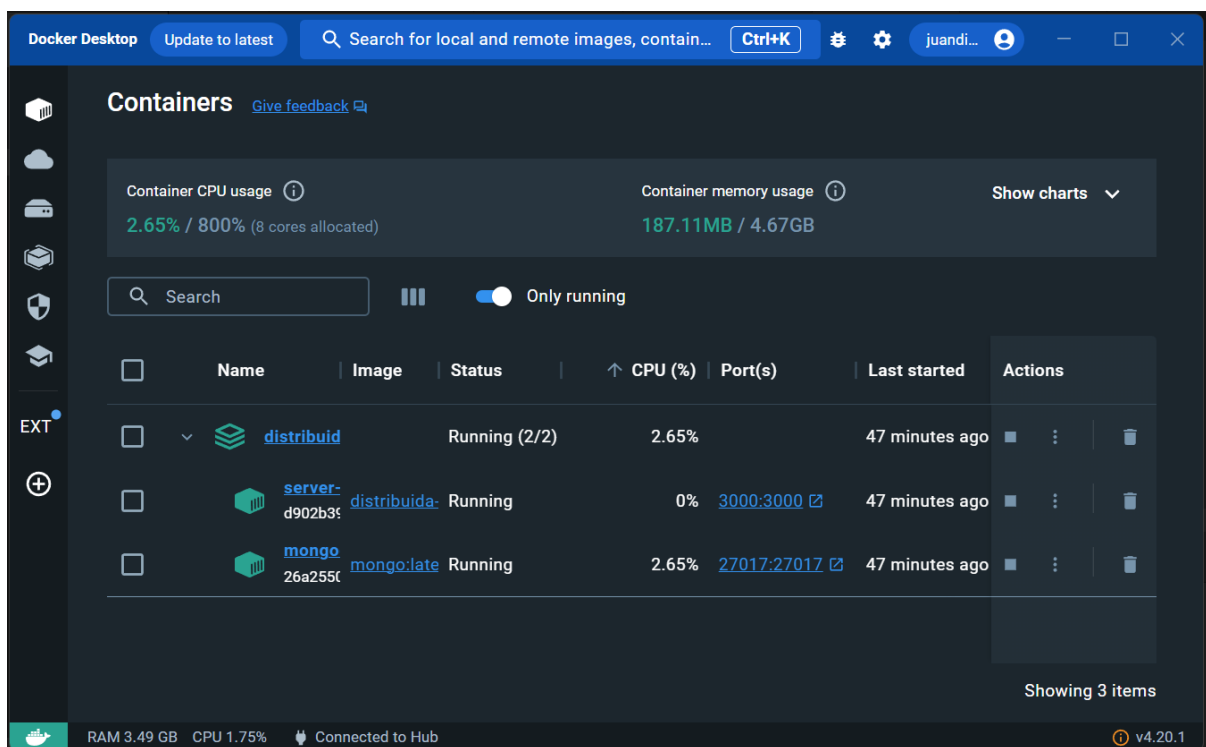
Uso de Sockets:

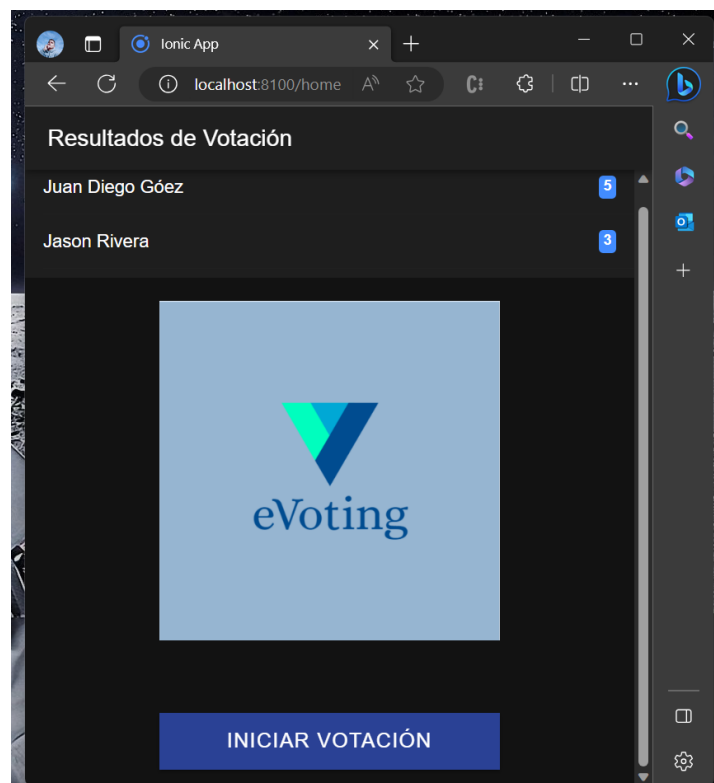
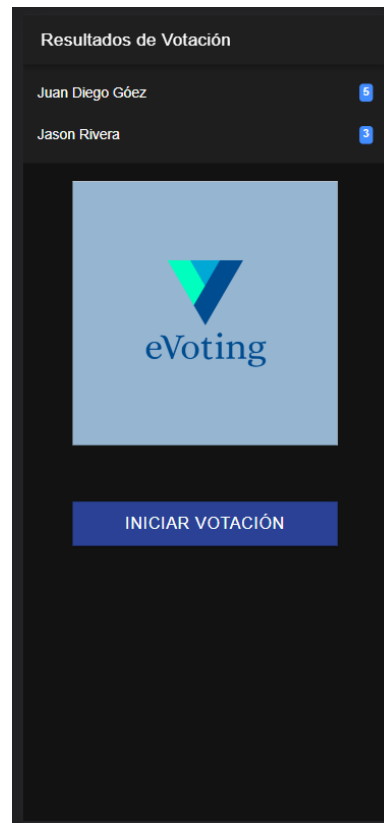
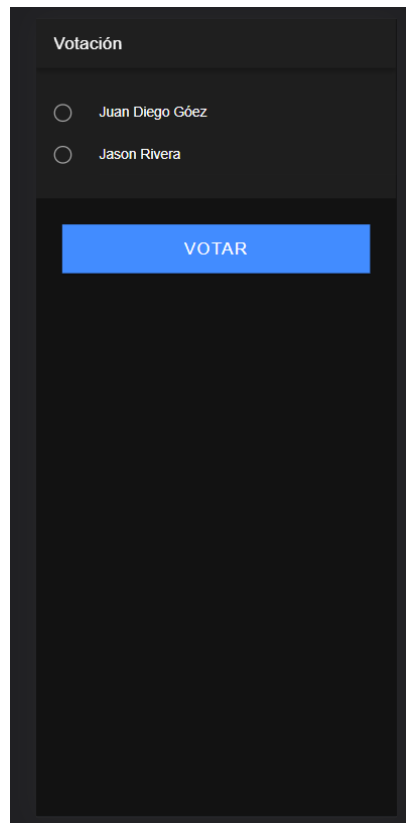
Los sockets son una tecnología fundamental en los sistemas distribuidos que permite la comunicación bidireccional entre diferentes componentes, incluso si están en diferentes máquinas o ubicaciones. En este sistema de votación, los sockets se utilizan para enviar actualizaciones en tiempo real sobre los votos emitidos. Cuando un usuario emite un voto, se envía a través de un socket al servidor, y luego el servidor notifica a todos los clientes conectados sobre el nuevo voto mediante el socket.

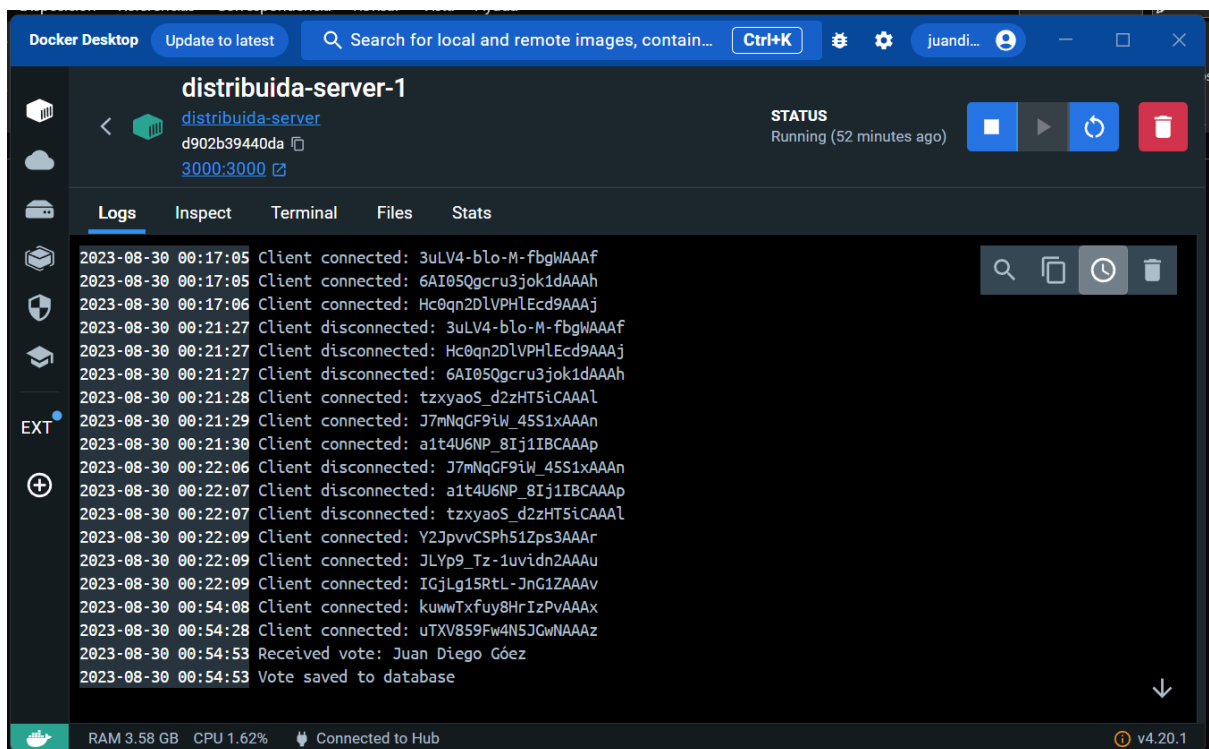
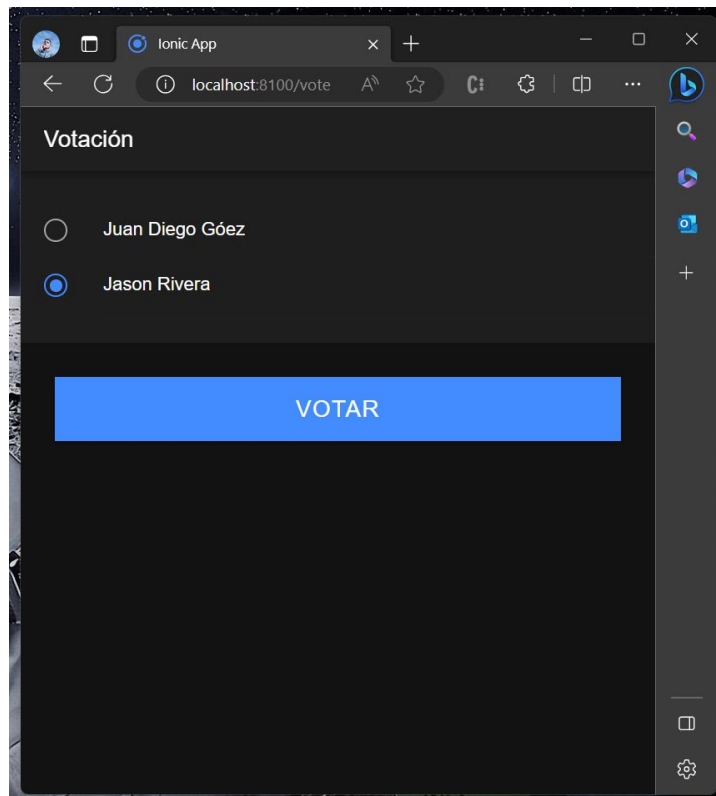
Uso de Docker:

Docker es una plataforma de contenedores que permite empacar una aplicación y todas sus dependencias en un "contenedor" que se puede ejecutar de manera consistente en diferentes entornos. En este sistema, Docker se utiliza para contener tanto la base de datos MongoDB como el servidor de votación. Esto garantiza que los entornos de desarrollo, prueba y producción sean coherentes y evita problemas de compatibilidad entre diferentes sistemas.

Capturas de funcionamiento:







localhost:27017 ...

Documents
voting_app.votes

My Queries

Databases

Search

- admin
- config
- local
- test
- voting_app
 - votes

voting_app.votes

Documents Aggregations Schema Indexes Validation

Filter ⌕ Type a query: { field: 'value' }

+ ADD DATA EXPORT DATA

```
{
  "_id": ObjectId('64eece97865d789b06dd1b13'),
  "option": "Juan Diego Góez",
  "__v": 0,
  "count": 5
}
```

```
{
  "_id": ObjectId('64eece9b865d789b06dd1b18'),
  "option": "Jason Rivera",
  "__v": 0,
  "count": 3
}
```

Configuración y despliegue:

Requisitos:

- Node.js
- Npm
- Angular
- Ionic
- Docker

Repositorio: <https://github.com/juandiego-gm/distribuidaEvoting.git>

Pasos:

1. Clonar el repositorio

2. Instalar las dependencias: Para esto en la terminal navega a la carpeta del servidor y de la aplicación moviel y ejecuta el comando “npm install” para instalar todas las dependencias necesarias.
3. Cambiar la URL del Socket.io: En la aplicación móvil, en el archivo app.module.ts y en home.page.ts, cambia la URL del socket.io para que coincida con la dirección IP o nombre de host del nuevo computador donde se está ejecutando el servidor. Por ejemplo: <http://192.168.0.100:3000>.
4. Iniciar Servidor y Base de datos: En la terminal, navega a la carpeta que contiene el archivo docker-compose.yml y ejecuta el comando docker-compose up para iniciar el contenedor del servidor y de MongoDB.
5. Compilar y Ejecutar la Aplicación Móvil: En la terminal, navega a la carpeta de la aplicación móvil y ejecuta el comando ionic serve para compilar y ejecutar la aplicación en el navegador.

Recuerda que, para probar en otro computador en la misma red local, necesitas asegurarte de que ambos computadores puedan comunicarse entre sí. Si estás probando en una red diferente, asegúrate de configurar las direcciones IP y los puertos correctamente para que la comunicación sea posible.

Conclusiones:

En conclusión, el sistema de votación distribuido aprovecha las ventajas de la descentralización, escalabilidad y tolerancia a fallos para brindar una experiencia confiable y eficiente a los usuarios. El uso de sockets permite la comunicación en tiempo real entre los componentes, y Docker garantiza la coherencia y portabilidad en diferentes entornos. Los sistemas distribuidos son esenciales en aplicaciones modernas para satisfacer las demandas de escalabilidad y disponibilidad, y esta aplicación de votación es un ejemplo claro de cómo se pueden aplicar estas tecnologías para lograr esos objetivos.