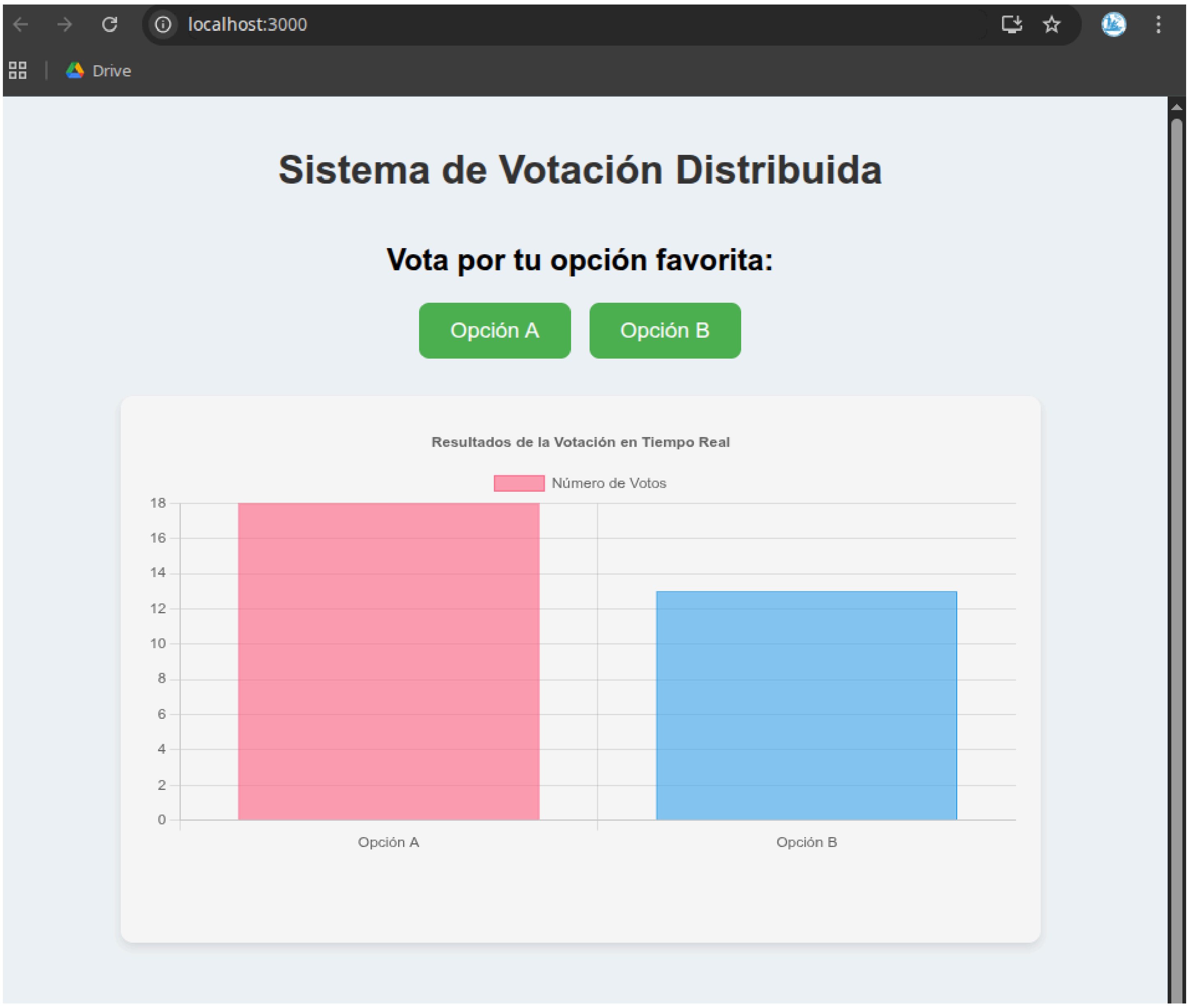
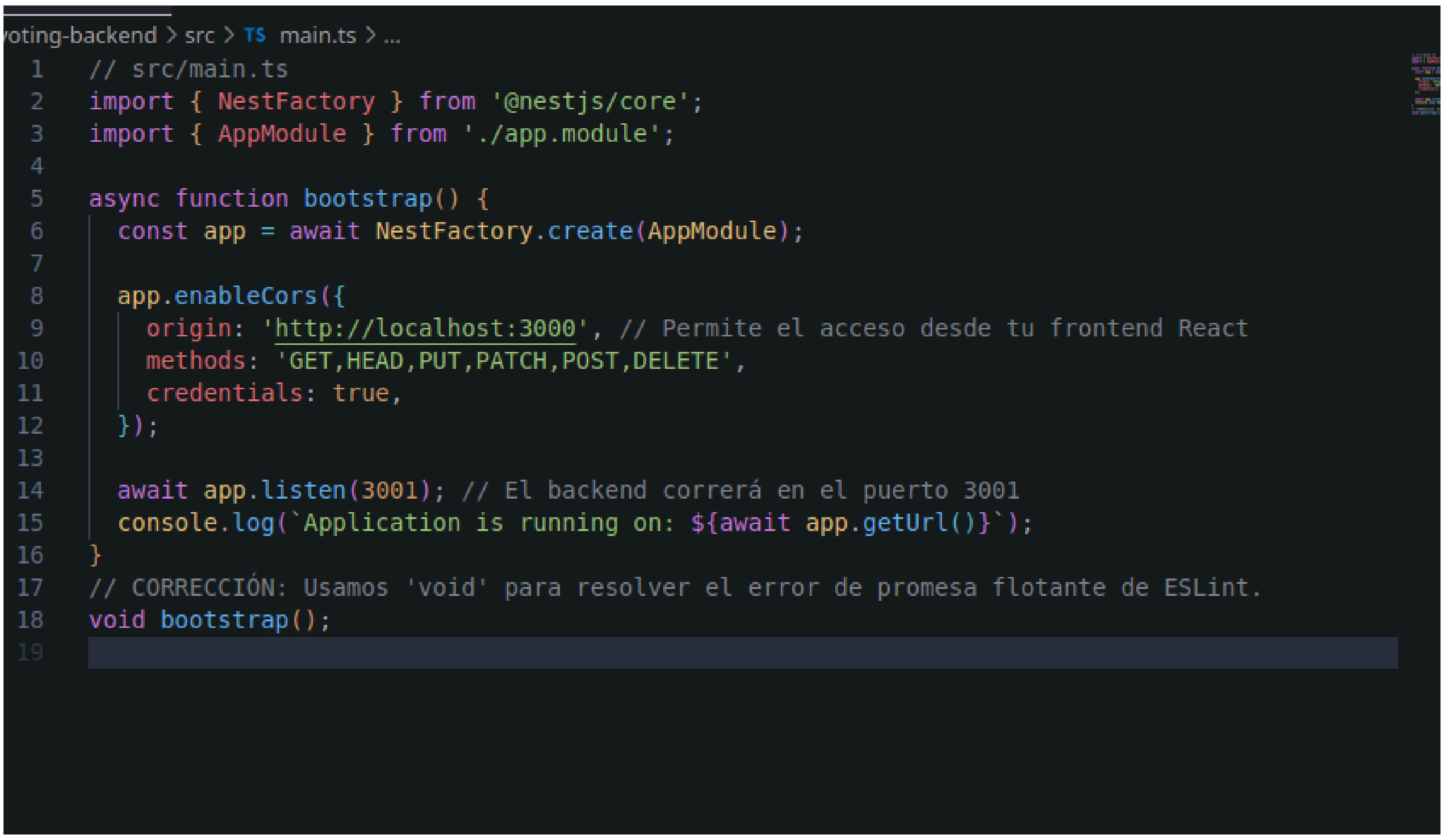


Sistema de Votación Distribuida

Vota por tu opción favorita:

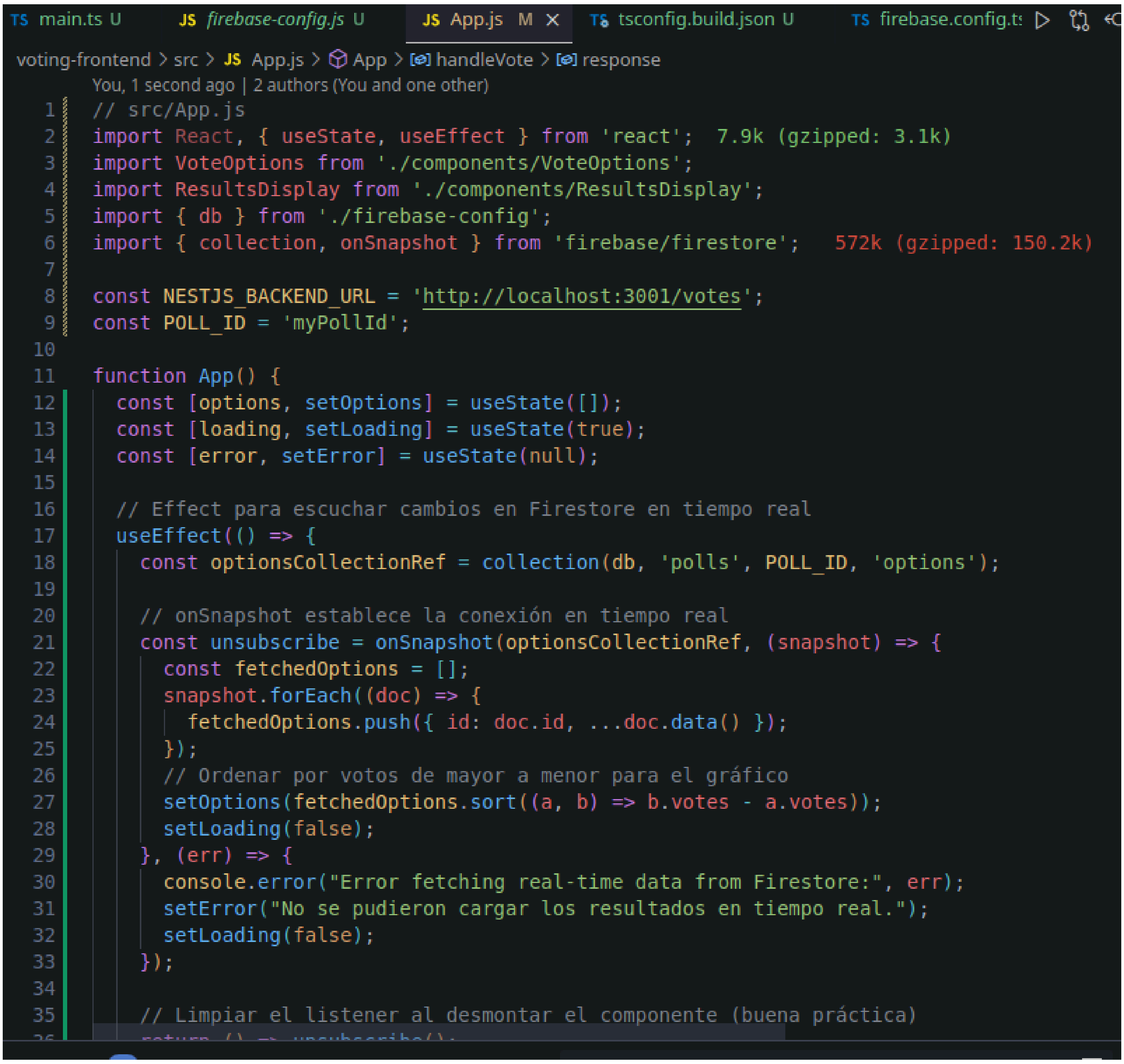


```
// src/main.ts
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app.module';

async function bootstrap() {
  const app = await NestFactory.create(AppModule);

  app.enableCors({
    origin: 'http://localhost:3000', // Permite el acceso desde tu frontend React
    methods: 'GET,HEAD,PUT,PATCH,POST,DELETE',
    credentials: true,
  });

  await app.listen(3001); // El backend correrá en el puerto 3001
  console.log(`Application is running on: ${await app.getUrl()}`);
}
// CORRECCIÓN: Usamos 'void' para resolver el error de promesa flotante de ESLint.
void bootstrap();
```


TS main.ts U
JS firebase-config.js U
JS App.js M ×
TS tsconfig.build.json U
TS firebase.config.ts
voting-frontend > src > JS App.js > App > handleVote > response
You, 1 second ago | 2 authors (You and one other)

```
// src/App.js
import React, { useState, useEffect } from 'react';  7.9k (gzipped: 3.1k)
import VoteOptions from './components/VoteOptions';
import ResultsDisplay from './components/ResultsDisplay';
import { db } from './firebase-config';
import { collection, onSnapshot } from 'firebase/firestore';   572k (gzipped: 150.2k)

const NESTJS_BACKEND_URL = 'http://localhost:3001/votes';
const POLL_ID = 'myPollId';

function App() {
  const [options, setOptions] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  // Effect para escuchar cambios en Firestore en tiempo real
  useEffect(() => {
    const optionsCollectionRef = collection(db, 'polls', POLL_ID, 'options');

    // onSnapshot establece la conexión en tiempo real
    const unsubscribe = onSnapshot(optionsCollectionRef, (snapshot) => {
      const fetchedOptions = [];
      snapshot.forEach((doc) => {
        fetchedOptions.push({ id: doc.id, ...doc.data() });
      });
      // Ordenar por votos de mayor a menor para el gráfico
      setOptions(fetchedOptions.sort((a, b) => b.votes - a.votes));
      setLoading(false);
    }, (err) => {
      console.error("Error fetching real-time data from Firestore:", err);
      setError("No se pudieron cargar los resultados en tiempo real.");
      setLoading(false);
    });

    // Limpiar el listener al desmontar el componente (buena práctica)
    return () => unsubscribe();
```