# Documento de arquitectura de software

## 1. Visión general

Tenemos una aplicación web para votación compuesta por 5 sistemas, como se detalla en la información de readme.md del repositorio https://github.com/dockersamples/example-voting-app:

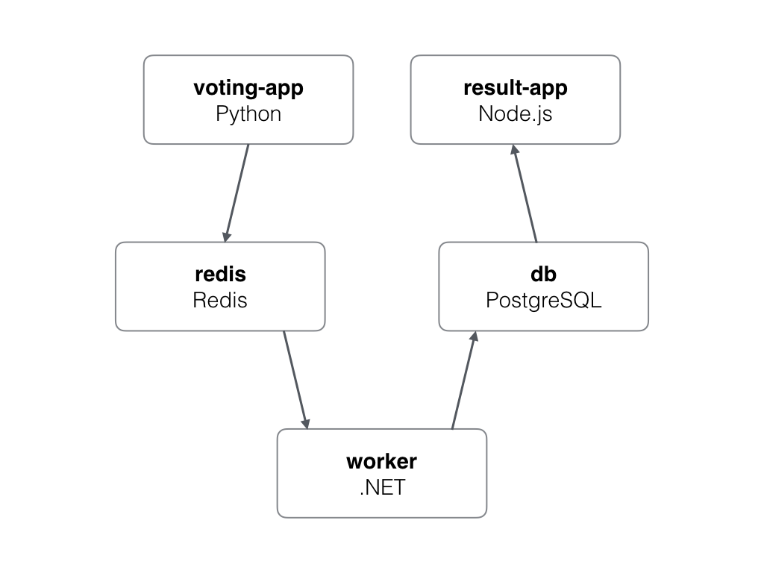
- Una aplicación web de Python que te permite votar entre dos opciones

- Una cola de Redis que recolecta nuevos votos

- Un trabajador .NET o Java que consume votos y los almacena en...

- Una base de datos de Postgres respaldada por un volumen de Docker

- Una aplicación web Node.js que muestra los resultados de la votación en tiempo real.



Se construyen a partir de docker-compose-javaworker-edited.yml, que posee la misma información que docker-compose-javaworker.yml, pero mapea db al puerto 5432 del host para poder ser accedido desde DBeaver.

### Diagramas de secuencia del comportamiento del sistema

Diagrama

Descripción generada automáticamente

Al emitir un voto el usuario realiza un post de su voto que se envía a voting-app (quien extrae el id de las cookies) y responde mostrando el valor del voto. Luego hace un push de un objeto JSON que contiene id y vote a Redis quien le asigna una clave y lo almacena.

Diagrama

Descripción generada automáticamente

El worker consume los datos de Redis ejecutando un bucle infinito que vacía la cola Redis (de forma FIFO) obteniendo los JSON con id y vote e insertándolos de la base de datos (en el caso de que la inserción falle se interpreta como que puede ser que ese id ya haya emitido un voto por lo que se procede a hacer un update).

Diagrama

Descripción generada automáticamente

Result-app hace una consulta a bd que retorna la cantidad de votos de a y la cantidad de votos de b y los muestra al usuario.

La estructura de la base de datos es la siguiente:

Tabla

Descripción generada automáticamente