

Laboratorio de Software Avanzado

iVoting - Fase 1

Grupo #3 • 18/02/2021

201503910 - Ariel Alejandro Bautista Méndez

201403905 - Javier Alberto Cabrera Puente

201314761 - Jorge David López Yool

200413020 - Dennis Josue Donis Mauricio

Descripción de la Aplicación

iVoting está pensado para ser un sistema que permita el voto electrónico, mediante un dispositivo móvil o página web que permita a los ciudadanos del país emitir su sufragio de una forma cómoda y segura, lo que ayudará a los connacionales a efectuar su voto desde el extranjero.



iVoting

Descripción Técnica

Base de datos

Para la base de datos se seleccionó mongo DB que contará con tres clusters diferentes. Uno para almacenar la información del usuario, otro para almacenar la información del voto, fecha de votación, y fotografías utilizadas para la votación, y en el tercero se llevarán los resultados de las elecciones en tiempo real.



Descripción Técnica

Back-end

El servidor será montado en nodeJS ya que es efectivo en el manejo de concurrencia de peticiones, además es liviano y amigable con la CPU, sin mencionar la facilidad de trabajar con los paquetes npm, documentación y gran comunidad que brinda soporte al framework.



Descripción Técnica

Front-end

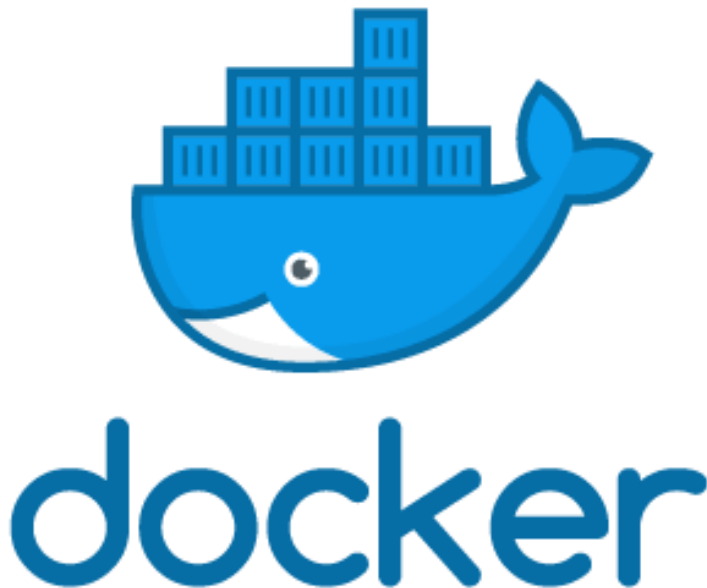
Debido al poco tiempo de desarrollo que se tiene, Angular es un framework que si bien, presenta una curva de aprendizaje bastante precipitada, una vez dominado, nos ahorramos tiempo en cuanto a la arquitectura y organización de la aplicación. Además al basarse en Typescript cualquier cambio que requiera la aplicación podrá implementarse de forma rápida y sin errores.



Descripción Técnica

Empaquetamiento de Software

Los contenedores Docker garantizan consistencia a lo largo del proceso de desarrollo y despliegue de una aplicación, estandarizando el entorno en el que se ejecuta. Una de las principales ventajas de una arquitectura basada en Docker es precisamente la estandarización. Docker permite desarrollos, compilaciones, pruebas y entornos de producciones repetibles y replicables y sin necesidad de compartir enormes archivos.



Descripción Técnica

Orquestador de Contenedores

Kubernetes es un sistema de código libre para la automatización del despliegue, ajuste de escala y manejo de aplicaciones en contenedores que fue originalmente diseñado por Google y donado a la Cloud Native Computing Foundation. Soporta diferentes entornos para la ejecución de contenedores, incluido Docker.



kubernetes

Descripción Técnica

Servidor de Nube

Google Cloud Platform no sólo nos permite acceder a una de las redes más grandes del mundo, sino que también garantiza que los clientes tengan menos tiempo de inactividad, ya que ahora podemos utilizar migraciones en vivo de máquinas virtuales.



Google Cloud Platform

Descripción Técnica

Gestor de Versiones

Al usar un gestor de versiones como GitHub, podemos trabajar en equipo más fácilmente, además de tener una mayor autonomía ya que cada desarrollador contará con un repositorio local, además que GitHub nos presenta su propio cliente de escritorio lo que agiliza la gestión del repositorio.

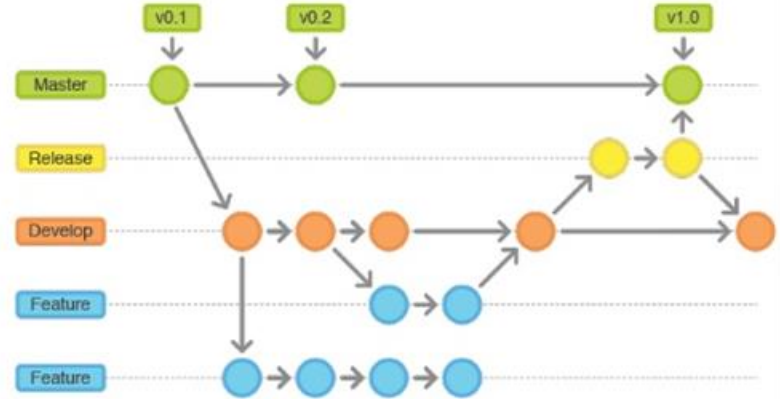


Descripción Técnica

Estrategia de Branching - Gitflow


Gitflow es un conjunto de técnicas de manejo de ramas basado en el modelo de ramificación de Vincent Driessen en donde las ramas Master y Develop son intocables, en Master se mantiene el código que está en producción o que se encuentra listo para llegar a esta etapa y en Develop se mantienen las *features* a incluir en una futura versión de la solución

Release Branches



Seguridad en la Aplicación

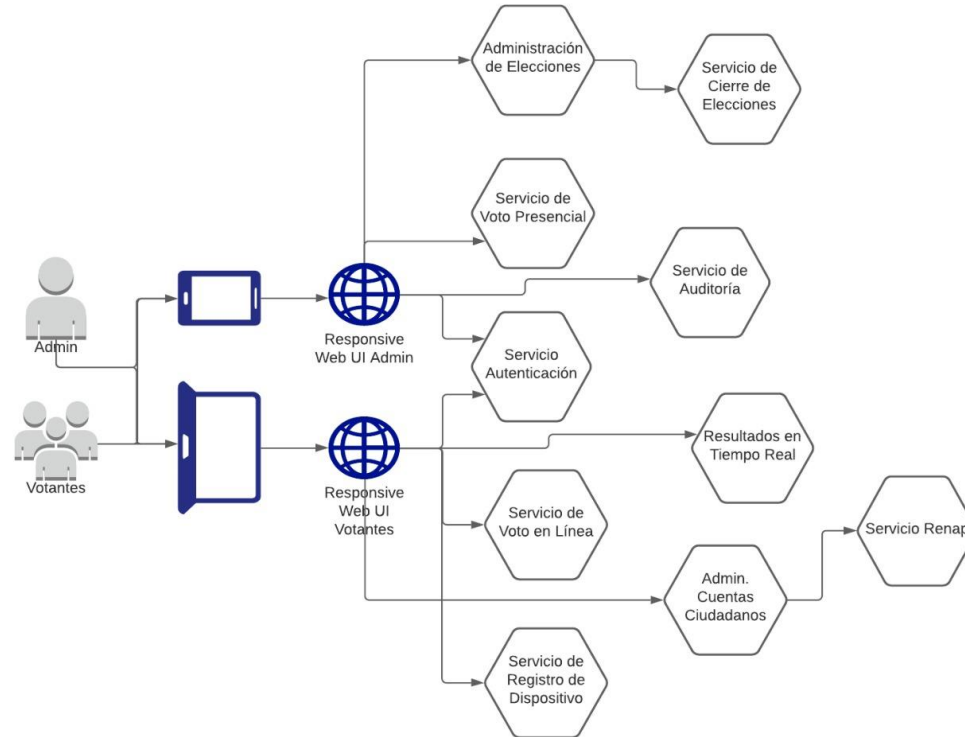
Mecanismos a Implementar

- Identificación mediante JSON Web Token (JWT) en cada invocación que se realice desde el cliente hacia el servidor.
 - Validación de permisos mediante JWT, es decir, si el cliente que realizó la petición tiene el nivel de privilegios adecuados para invocar un determinado servicio.
 - Parametrización de las consultas SQL en el servidor para evitar SQL Injection.
 - Validación del dispositivo desde donde se realiza la petición mediante User-Agent
 - Cifrado de información sensible en la base de datos, como passwords, y secret keys.
 - Implementación de Amazon Rekognition para validar la identidad de la persona al emitir el sufragio.
 - Límite de tiempo de las sesiones.
- 

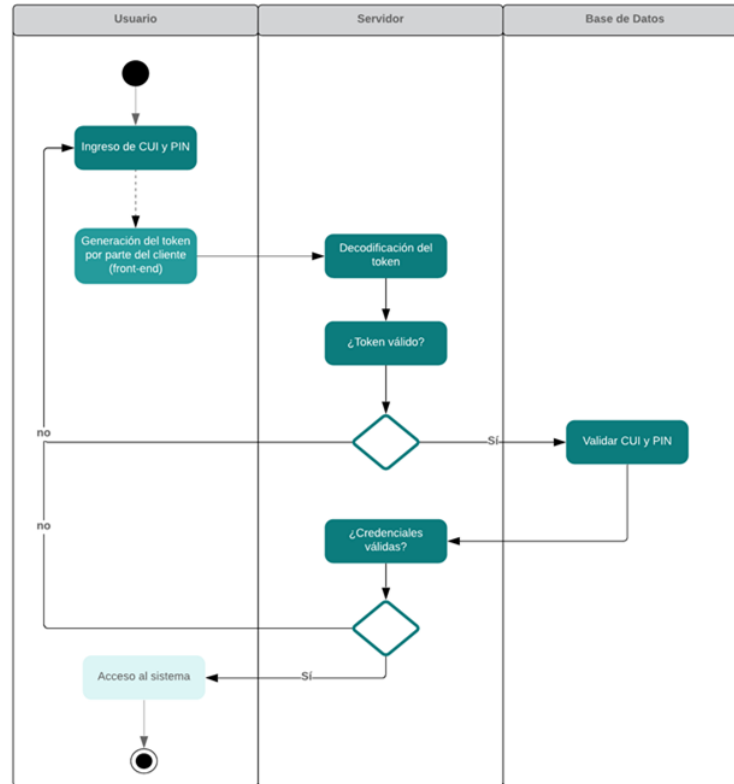


Arquitectura de la Solución

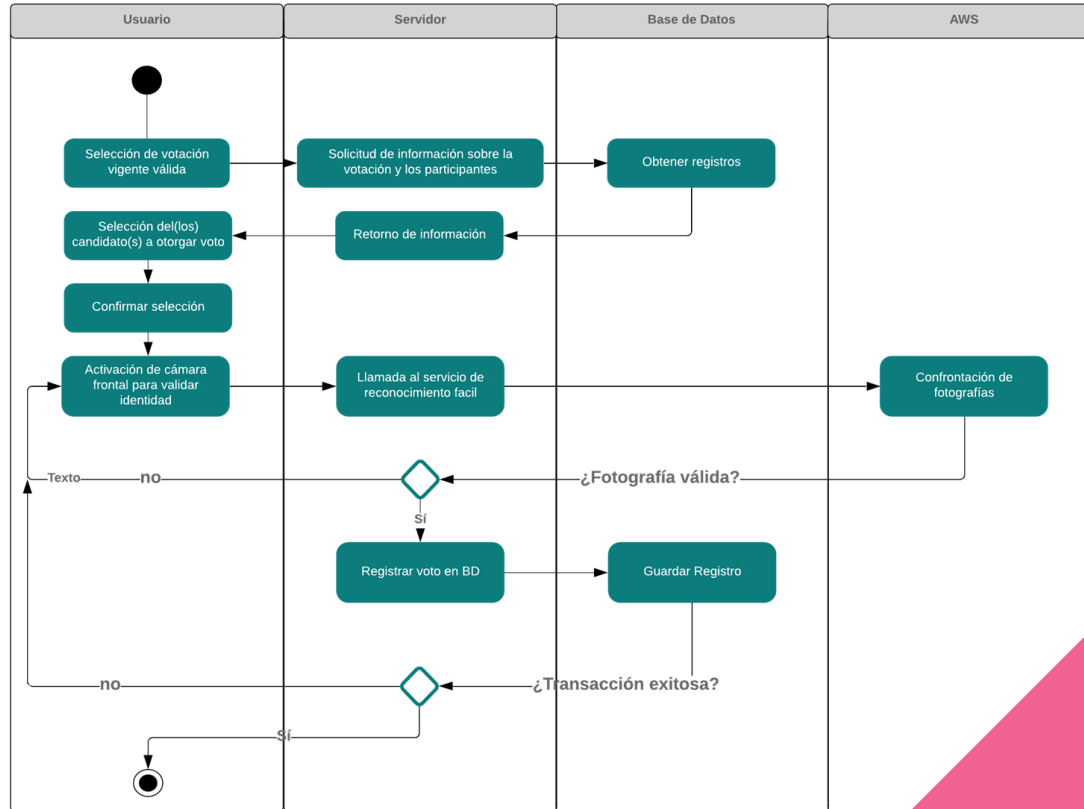
Comunicación Entre Servicios



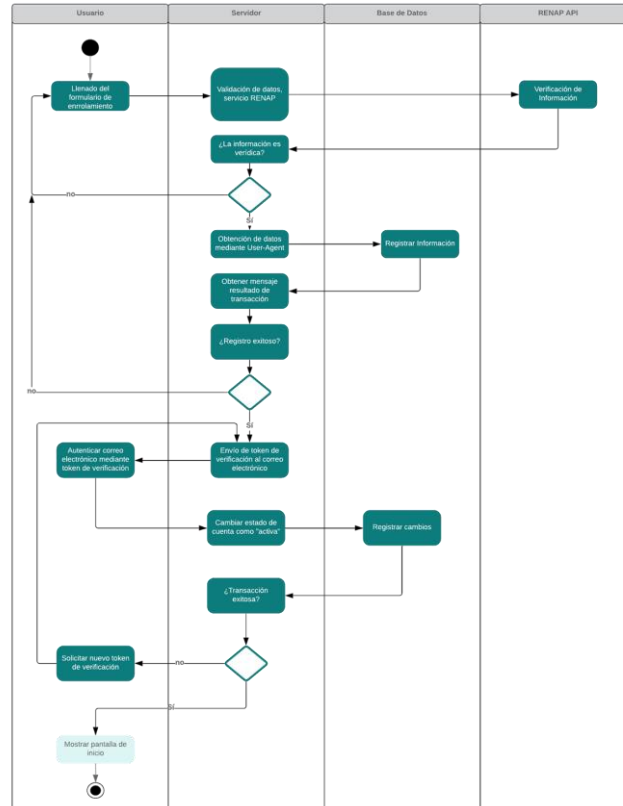
Autenticación de Usuarios - Diagrama de Actividades



Emisión de Voto - Diagrama de Actividades



Sistema de Registro Ciudadano - Diagrama de Actividades



Microservicios

Microservicios Identificados

1. Servicio de Administración de Elecciones (Creación de Elecciones)
2. Servicio de Administración de Cuentas Ciudadanos (CRUD de Ciudadanos)
3. Servicio de Registro de Dispositivo
4. Servicio de Autenticación
5. Servicio de Voto en Línea
6. Servicio de Voto Presencial
7. Servicio de Resultados en Tiempo Real
8. Servicio de Auditoría
9. Servicio de Cierre de Elecciones

Objetivos Para la 2da. Fase

1. Pipeline para el despliegue automático de cambios.
 2. Modelo de datos a utilizar.
 3. Ejecución automática de pruebas unitarias.
 4. Implementación de los microservicios en contenedores e implementación del sistema de orquestación.
 5. Pruebas de endpoints.
-