Documento de Diseño y Arquitectura del Sistema

GanTrack v0.1.0 **Rev-000**

19 de noviembre de 2019

Índice

1. Historial de Revisión

Fecha	Versión	Autor	Descripción
19 de noviembre de 2019	v0.1.0	DFCH	Primer borrador del documento

2. Visión

2.1. Propósito

Este documento describe a sobre manera la solución técnica para el proyecto GanTrack. Dicha solución fue concebida con el objetivo de abordar todos los requerimientos descritos en el Documento de Requerimientos de Negocio de GanTrack. Aquí se describen los componentes más importantes para llevar a cabo la solución, las dependencias entre ellos y cómo es que trabajan en conjunto.

El presente documento tiene el propósito de:

- Mostrar las desiciones a nivel técnico respecto a la arquitectura.
- Permitir la integración de nuevos desarrolladores de manera efectiva y más eficiente.
- Habilitar una transición fluida entre operaciones.

Este es un documento vivo, se espera que los arquitectos lo actualizarán conforme el proyecto avance y el sistema sea promovido a la etapa de producción.

2.2. Contexto de Negocio

ColTrack pretende ofrecer una herramienta de monitoreo para colmenas, de manera que sea posible el detectar cuando una de estas es abierta de manera no autorizada y/o movida fuera de una zona apícola predeterminada sin autorización.

2.3. Alcances

Las áreas cubiertas en este documento incluyen el diseño de las vistas del sistema y los componentes incluidos, así como una vista general de la arquitectura básica que éste debe tener de acuerdo a los requerimientos.

2.4. Fuera de Alcance

No se cubre en este documento la descripción detallada de las tecnologías utilizadas para desarrollar la solución o detalles específicos de la programación en general de los componentes para llevar a cabo con éxito los requerimientos solicitados.

3. Lista de Abreviaciones

Abreviación	Descripción
IoT	Internet of Things (Internet de las cosas)
MVC	Modelo Vista Controlador

4. Consideraciones de Diseño

4.1. Supuestos

- Se utilizará un servidor virtual de Linux.
- El sistema operativo por defecto será Ubuntu 16.04.
- Los usuarios serán dueños o administradores de un conglomerado de colmenas.
- Además del rastreo y monitoreo de apertura de las colmenas, será posible agregar otros parámetros de medición que sean del interés al usuario y podrán ser diferentes de acuerdo a las necesidades del usuario.
- El usuario cuenta con una conexión a internet.

4.2. Restricciones

- Los datos de los dispositivos son obtenidos desde un servidor externo administrado por la empresa Sigfox, por lo que se depende de éste servicio para el monitoreo de dispositivos.
- Es necesario que las aplicaciones clientes y el servidor que administra los dispositivos cuenten con conexión a internet para ser utilizados, dada la restricción anterior. Esto debido a que los datos son enviados desde la nube Sigfox bajo el protocolo http desde un servidor remoto.

4.3. Riesgos

- No se cuenta con los servidores físicos propios para el despliegue, prueba y desarrollo del proyecto, sólo se cuenta con servidores virtuales anidados dentro de un servidor propio de la Universidad, por lo que no es posible asegurar la disponibilidad del sistema en todo momento.
- No es posible modificar o crear los servidores de nombres por cuenta de los desarrolladores, así que las aplicaciones deberán trabajar bajo el nombre de una IP, lo que limita el número de aplicaciones que pueden ser publicadas desde el servidor.
- El ancho de banda disponible, así como su consecuente latencia y velocidad de conexión depende del prestador del servicio de internet contratado por la Universidad y el contrato adquirido por ésta y el prestador del servicio.

4.4. Estrategias

Se usará el enfoque de servidores con la tecnología Express-js y Node-js, dado que proveen un framework de servidores web con características convenientes para aplicaciones de internet de las cosas (IoT) como lo son, un mayor rendimiento en el procesamiento de los datos y una curva de aprendizaje menor, dado que se trata de el lenguaje de programación Javascript, usado también en el diseño de las páginas web y el procesado de datos en las mismas.

Por otra parte, se utilizará como servidor de proxy inverso, o gateway interno, al motor de Nginx, ya que permitirá que se puedan utilizar diferentes aplicaciones web dentro de un mismo servidor sin la necesidad de abrir un puerto por cada aplicación, además de que es recomendado también para aplicaciones que procesan datos en tiempo real, debido al control del rendimiento que ofrece por medio de sus distintas configuraciones [1].

Se tomó en cuenta el uso de Apache como servidor proxy, pero distintas pruebas han demostrado que su rendimiento es menor al de Nginx al procesar datos en tiempo real [1].

Entonces, la distribución del sistema sería: un servidor proxy inverso como manejador de tráfico entre aplicaciones, un servidor o varios servidores de aplicación web, un servidor de rececpión de datos que los dispositivos remotos envíen y un servidor de base de datos.

Con las decisiones explicadas anteriormente, se pretende que el sistema sea escalable desde su concepción y permita que sus características varíen de acuerdo a los requerimientos de cada cliente sin que estos afectne el diseño del sistema en sus componentes críticos.

5. Arquitectura

5.1. Vista General del sistema

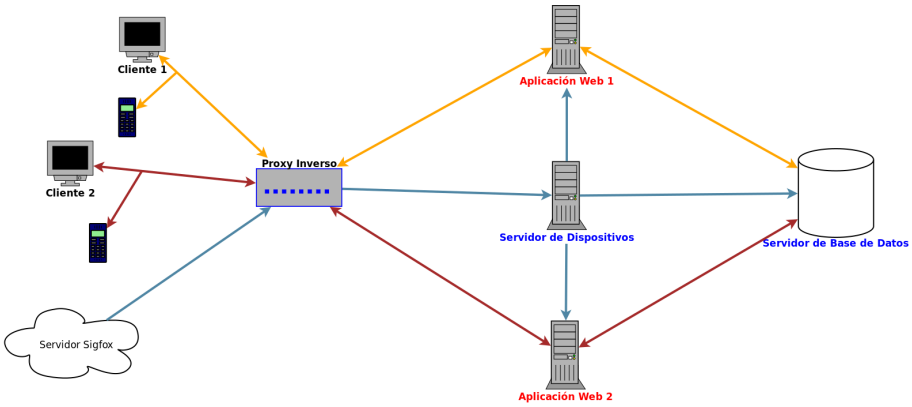

Figura 1: Vista General del Sistema

La arquitectura del sistema, mostrada en la Figura 1 se basa en el enfoque Modelo-Vista-Controlador (MVC) con el objetivo de dividir las funcionalidades y tratar los datos de manera más eficiente. Siendo el Modelo la base de Datos operada por el servicio de MongoDB, la vista son los Servidores de Aplicación Web, que renderizan las páginas web antes de mostrarlas al cliente y el Controlador el Servidor de Dispositivos es encargado de procesar en tiempo real los datos en que los dispositivos pudiesen enviar. Se espera que este último servidor tenga una comunicación abierta y constante hacia el servidor Sigfox, intermediario entre los dispositivos y el propio controlador, así también manteniendo los datos actualizados en todo momento.

5.2. Vista Lógica

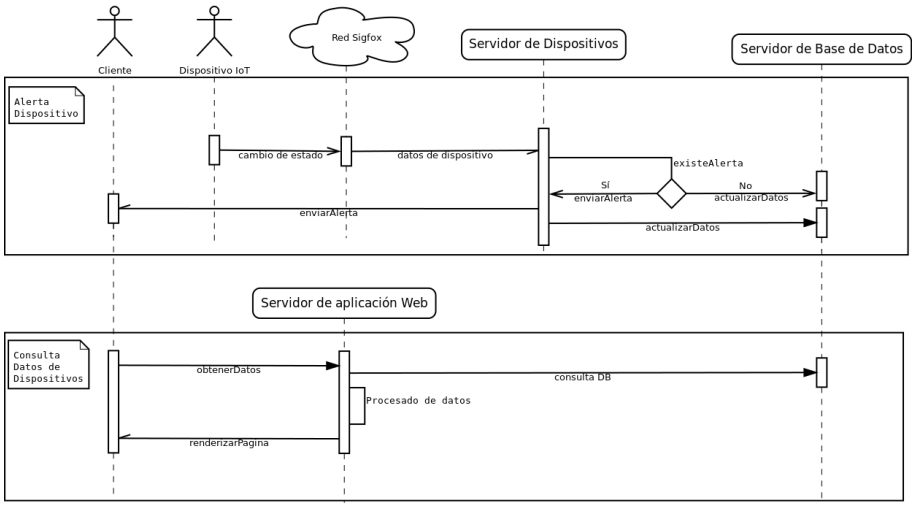

Figura 2: Diagrama de flujo para los eventos principales en el sistema

En este diagrama (Figura 2) se muestran los eventos principales que se tomaron en cuenta para el diseño y desarrollo del sistema, donde el manejo de alarmas en tiempo real puede utilizarse como base para un subsecuente monitoreo en tiempo real de cualquier dato que los dispositivos IoT pudieran medir sin perder de vista que debe realizarse un registro de dichos datos para tener una referencia a futuro o un procesado estadístico de estos.

Es así que el Servidor de aplicación Web sería el encargado de procesar dichos datos y renderizar la representación gráfica de éstos antes de enviarlos al cliente tomando en cuenta los requerimientos de cada usuario final sobre la presentación, procesamiento y estadísticas con los que, en su momento, quisieran contar.

5.3. Vista de Componentes

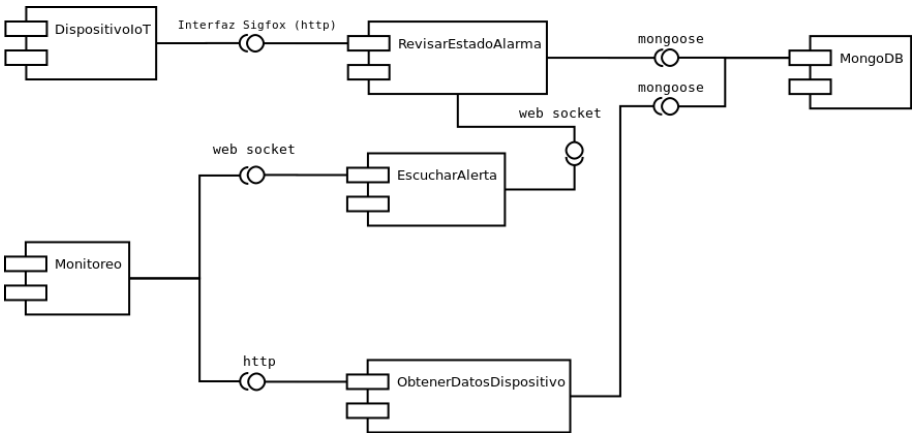

Figura 3: Vista de las relaciones entre los componentes del sistema

En la Figura 3 se representan distintos componentes del sistema, así como sus interacciones, y se muestran de manera general las interfaces requeridas para el funcionamiento de éstos.

6. Diseño de Base de Datos

En esta sección se muestra el diseño de la base de datos propuesta para dar solución a los requerimientos y la relación numérica entre cada modelo de datos tomado en cuenta.

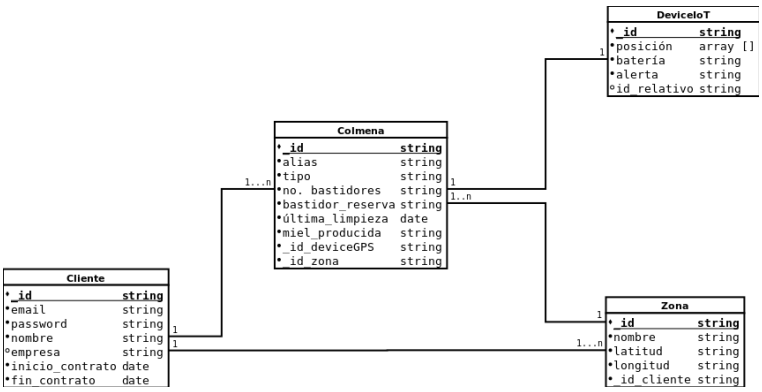

Figura 4: Modelo de base de datos para GanTrack

Como es posible observar en la Figura 4, el elemento crítico para procesar los datos y, por ende, donde se concentrarán los requerimientos del sistema, es el modelo de Colmena. Dicho modelo estará asociado a un solo DeviceIoT, el cuál informará de cambios en el estado cada cierto tiempo o dado un evento determinado, para este modelo se tomó en cuenta sólo el cambio de posición y una alarma.

Cada Colmena estará asociada a una Zona delimitada por cierta longitud y latitud, y que será creada y administrada por un cliente registrado, el cuál puede tener una o varias colmenas en una o varias Zonas predeterminadas. Dicho cliente estará sujeto a un contrato de operación, por lo que se toma en cuenta la fecha de inicio y final de tal contrato.

Cabe señalar que los tipos de datos son en su mayoria *string* dado que se utilizará una base de datos en base a documentos, la cuál procesa los datos como cadenas de texto, por lo que se facilitará la operación no tener que convertir los datos a otro tipo.

Referencias

[1] Nginx vs Apache – which is the best web server?. (2019). Recuperado el 19 de noviembre de 2019, desde https://www.plesk.com/blog/various/nginx-vs-apache-which-is-the-best-web-server/

[2] What is Dynamic Content? - Articles for Developers Building High Performance Systems. (2019). Recuperado el 19 de noviembre de 2019, desde https://blog.stackpath.com/glossary-dynamic-content/