VII. Alcance

Define el alcance del proyecto:

* Obtención de datos como consumo de los dispositivos del hogar.
* Control de encendido a distancia a través de la web
* Facturación por frecuencia de tiempo
* Visualización de datos en tiempo real a través del portal web.

. Todo proyecto de grado tiene ciertas limitaciones por temas de conocimiento, recursos, tiempo y otros factores. En esta sección se indican cuáles fueron los límites acordados el comité (y asesor) de grado.

XII. Metodología

¿Cómo se resolvió el problema?

* A través del backend montando en NodeJS, se exponen 7 API’s relacionado a los objetos de clientes, localidades, plugs, dispositivos, consumos, login y facturación. Por estos API’s se logran realizar todas las actividades concernientes a los objetos antes mencionados tales como lectura, creación, actualización y eliminación, también conocidos como operaciones CRUD.
* En el frontend montado en ReactJS interactuamos y operamos la información suministrada por los API’s de los objetos, así como también obtener la información en tiempo real de los consumos.

¿Qué se implementó y cómo se llevó a cabo?

* Para el backend utilizamos la tecnología de NodeJS, para crear API’s para los principales objetos del sistema. Del mismo modo en el backend utilizamos los siguientes frameworks:
  + Express. Utilizado para la gestión de peticiones en diferentes rutas del backend, así como también para establecer los ajustes en el servidor de NodeJS.
  + JsonWebToken (JWT). Utilizado para crear tokens para cada usuario al momento al iniciar sesión, sin el token los usuarios no podrían iniciar sesión en el sistema.
  + Bcrypt. Utilizado para encriptar las contraseñas al momento del usuario registrarse.
  + MomentJS. Utilizado para darle formato a las fechas y horas obtenidas desde el servidor.
  + Socket.io. Utilizado para recibir y enviar en tiempo real informaciones de los consumos consultados en la base de datos hacia el frontend.
* Para el frontend utilizamos la tecnología de ReactJS, para gestionar y manipular la información obtenida desde el backend. Del mismo modo en el backend utilizamos los siguientes frameworks:
  + Axios. Utilizado para gestionar las peticiones desde el frontend hacia el backend.
  + JsonWebToken (JWT). Utilizado para desencriptar el token, y obtener las informaciones básicas del usuario (id y email).
  + Bootstrap. Utilizado para gestionar el diseño y estructura de los elementos HTML del frontend.
  + Reactstrap. Utilizado para manipular los elementos JSX del frontend.
  + ChartJS. Utilizado para crear los gráficos de barra con la información obtenida.
  + Socket.io. Utilizado para recibir y enviar en tiempo real informaciones del backend.

XIII. Recursos

Enumerar los recursos (hardware, software, licencias, suscripciones, etc.)

* NodeJS.
* Express
* MomentJS.
* Bcrypt.
* JsonWebToken (JWT).
* Socket.io.
* ReactJS.
* Axios.
* Bootstrap.
* Reactstrap.
* ChartJS.

XIV. Resultados encontrados en la experimentación

El título lo dice todo. Solo los relevantes para el problema en cuestión o para el funcionamiento integral de la solución, no todos.

XVII. Problemas presentados

* Lógica para mostrar la información en tiempo real (la data se inserta cada 30 segundos, no cada 1 segundo por performance de la VM).