**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 2**

-----------------------------

Ảnh có chứa Đồ họa, thiết kế đồ họa, Phông chữ, biểu tượng

Mô tả được tạo tự động

**Báo cáo cuối kỳ**

**Môn: IoT và ứng dụng**

**Đề tài:**

**“XÂY DỰNG NODE-RED DASHBOARD SỬ DỤNG DỮ LIỆU CẢM BIẾN OPENEEW”**

**Giảng viên hướng dẫn: Đàm Minh Lịnh**

**Lớp: D20CQCNPM01-N**

**Nhóm sinh viên thực hiện: Huỳnh Gia Hưng - N20DCCN024**

**Đậu Văn Hoàng - N20DCCN017**

**Mục lục**

[**I. Tổng quan. 3**](#_Toc155319542)

[**1. Vấn đề về động đất. 3**](#_Toc155319543)

[**2. Tìm hiểu chung về OpenEEW data. 3**](#_Toc155319544)

[**II. Mô hình đề xuất cho ứng dụng. 4**](#_Toc155319545)

[**1. Node-Red 4**](#_Toc155319546)

[1.1. Node-Red là gì? 4](#_Toc155319547)

[1.2. Các tính năng của Node-RED. 4](#_Toc155319548)

[1.3. Có thể cài đặt Node-RED ở đâu? 4](#_Toc155319549)

[1.4. Ứng dụng của Node-RED. 4](#_Toc155319550)

[1.5. Cài đặt Node-Red trên Windowns. 4](#_Toc155319551)

[**2. NodeJS 4**](#_Toc155319552)

[2.1. NodeJS là gì? 4](#_Toc155319553)

[2.2. Ứng dụng của NodeJS 4](#_Toc155319554)

[2.3. Ưu nhược điểm của NodeJS 4](#_Toc155319555)

[**III. Triển khai, trải nghiệm và ứng dụng của phần mềm. 7**](#_Toc155319556)

[**1. Dữ liệu OpenEEW. 7**](#_Toc155319557)

[1.1. Truy cập dữ liệu OpenEEW trên AWS. 7](#_Toc155319558)

[1.2. Trường dữ liệu. 7](#_Toc155319559)

[1.3. Cách các bản ghi được gán cho một tập tin. 8](#_Toc155319560)

[1.4. Siêu dữ liệu thiết bị 8](#_Toc155319561)

[**2. Khai triển ứng dụng 9**](#_Toc155319562)

[**IV. Kết luận và hướng phát triển. 20**](#_Toc155319563)

[**Tư liệu tham khảo 21**](#_Toc155319564)

1. **Tổng quan.**
2. **Vấn đề về động đất.**

Động đất hay Địa chấn (Earthquake) là sự rung chuyển trên bề mặt Trái Đất do kết quả của sự giải phóng năng lượng bất ngờ ở [lớp vỏ Trái Đất](https://vi.wikipedia.org/wiki/L%E1%BB%9Bp_v%E1%BB%8F_(%C4%91%E1%BB%8Ba_ch%E1%BA%A5t)) và phát sinh ra [sóng địa chấn](https://vi.wikipedia.org/wiki/S%C3%B3ng_%C4%91%E1%BB%8Ba_ch%E1%BA%A5n). Hoạt động địa chấn của một khu vực là tần suất, loại và kích thước của trận động đất trải qua trong một khoảng thời gian.

Những tại hại của động đất:

• Tác động trực tiếp của các trận động đất là rung cuộn mặt đất, gây ra hiện tượng nứt vỡ, làm sụp đổ các công trình xây dựng, gây sạt lở đất, lở tuyết. Mức độ nghiêm trọng của nó dựa trên cường độ, khoảng cách tính từ chấn tâm, và các điều kiện về địa chất, địa mạo tại nơi bị ảnh hưởng.

• Động đất cũng thường gây ra hỏa hoạn khi chúng phá hủy các đường dây điện và các đường ống khí.

• Các trận động đất xảy ra dưới đáy biển có thể gây ra lở đất hay biến dạng đáy biển, làm phát sinh sóng thần (những đợt sóng lớn tràn qua các đại dương rồi đổ bộ vào đất liền). Đôi khi động đất còn khiến núi lửa hoạt động, thậm chí là những núi lửa đã tắt từ lâu...

* Động đất gây ra rất nhiều thiệt hại về người và tài sản của nhân loại nên chúng ta cần có những biện pháp để hạn chế những thiệt hại của động đất. Cách tốt nhất là tạo ra hệ thống cảnh báo động đất sớm để có thể giảm được thiệt hại cho người dân. Nên chúng em sẽ sử dụng dữ liệu cảm biến openEEW để tạo ra hệ thống cảnh báo động đất sớm.

1. **Tìm hiểu chung về OpenEEW data.**

Hệ thống cảnh báo sớm động đất (EEW) là một tập hợp các công nghệ phát hiện và mô tả đặc điểm của một trận động đất lớn, nhanh chóng gửi thông tin đến các cộng đồng lân cận trước khi động đất xảy ra để họ có thể chuẩn bị và hành động phù hợp.

Tất cả các dự án OpenEEW đều yêu cầu các thành phần sau:

• Một mạng lưới các cảm biến OpenEEW .

• Một hệ thống phát hiện đang chạy trên một máy chủ .

• Một phương pháp gửi cảnh báo . Ví dụ: đối với tài khoản Twitter, ứng dụng di động hoặc thiết bị IoT.

1. **Mô hình đề xuất cho ứng dụng.**

Mô hình ứng dụng sử dụng công cụ mã nguồn mở Node-Red và NodeJS.

1. **Node-Red**
   1. **Node-Red là gì?**

**Node-RED**là một công cụ mã nguồn mở và trực quan được sử dụng để xây dựng các luồng làm việc (workflows) và ứng dụng Internet of Things (IoT). Nó cung cấp một giao diện đồ họa dựa trên trình duyệt web, cho phép người dùng kết nối các nút (node) với nhau để xử lý dữ liệu và tương tác với các thiết bị và dịch vụ khác nhau.

**Node-RED**được xây dựng trên nền tảng Node.js và sử dụng trình duyệt web để tạo ra một giao diện dễ sử dụng. Người dùng có thể kéo và thả các nút từ thư viện có sẵn để tạo ra luồng làm việc theo ý muốn. Các nút có thể thực hiện các nhiệm vụ khác nhau, bao gồm xử lý dữ liệu, kết nối và tương tác với các dịch vụ web, cơ sở dữ liệu, thiết bị IoT và nhiều hơn nữa.

**Node-RED**có một cộng đồng rộng lớn và có sẵn nhiều bộ nút mở rộng, cho phép người dùng tương tác với các nền tảng và dịch vụ phổ biến như MQTT, HTTP, MySQL, MongoDB, Twitter, Raspberry Pi, Arduino và nhiều hơn nữa. Công cụ này rất linh hoạt và phù hợp cho việc phát triển các ứng dụng IoT, tự động hóa và quản lý dữ liệu.

* 1. **Các tính năng của Node-RED.**

Node-RED có nhiều tính năng hữu ích và mạnh mẽ để xây dựng và quản lý các luồng làm việc (workflows) và ứng dụng IoT. Dưới đây là một số tính năng chính của Node-RED:

* **Giao diện trực quan:**Node-RED cung cấp một giao diện đồ họa dễ sử dụng dựa trên trình duyệt web. Người dùng có thể kéo và thả các nút để tạo và kết nối các luồng làm việc một cách trực quan.
* Các nút và luồng đa dạng: Node-RED đi kèm với một bộ sưu tập nút mở rộng lớn, cho phép bạn tương tác với các thiết bị, dịch vụ và giao thức khác nhau. Các nút này bao gồm MQTT, HTTP, cơ sở dữ liệu, xử lý dữ liệu, gửi email, gửi tin nhắn và nhiều hơn nữa. Bạn cũng có thể tạo các nút tùy chỉnh của riêng mình.
* **Tích hợp dịch vụ web:**Node-RED cho phép bạn tương tác với các dịch vụ web phổ biến như Twitter, Facebook, Dropbox và nhiều dịch vụ khác. Bạn có thể gửi yêu cầu API, nhận dữ liệu từ các dịch vụ này và thực hiện các hành động liên quan.
* **Tích hợp IoT:** Node-RED được phát triển đặc biệt cho việc xây dựng ứng dụng IoT. Bạn có thể tương tác với các thiết bị như Raspberry Pi, Arduino, các cảm biến và các nền tảng IoT khác. Node-RED cung cấp các nút và giao thức như MQTT, CoAP và Modbus để liên kết và thu thập dữ liệu từ các thiết bị IoT.
* **Xử lý dữ liệu linh hoạt:**Node-RED cho phép bạn xử lý, biến đổi và lọc dữ liệu theo ý muốn. Bạn có thể sử dụng các nút có sẵn để thực hiện các hoạt động như chuyển đổi định dạng, phân tích văn bản, tính toán, lưu trữ và đồng bộ dữ liệu.
* **Giao tiếp thời gian thực:**Node-RED hỗ trợ giao tiếp thời gian thực và khả năng phát hiện sự kiện. Bạn có thể tạo các luồng làm việc để theo dõi, giám sát và phản ứng.
  1. **Có thể cài đặt Node-RED ở đâu?**

Node-RED có thể được sử dụng ở nhiều định dạng và môi trường khác nhau. Dưới đây là một số ví dụ về nơi mà Node-RED có thể được triển khai:

* **Máy tính cá nhân:**Bạn có thể cài đặt Node-RED trực tiếp trên máy tính cá nhân của bạn, bao gồm hệ điều hành Windows, macOS và Linux. Điều này cho phép bạn xây dựng và chạy các luồng làm việc trên máy tính để tự động hóa các quy trình, xử lý dữ liệu và tương tác với các dịch vụ khác nhau.
* **Máy chủ:**Node-RED có thể được triển khai trên máy chủ để xây dựng ứng dụng và dịch vụ phức tạp hơn. Bạn có thể cài đặt Node-RED trên các máy chủ vật lý hoặc máy chủ ảo, như máy chủ Linux hoặc Windows Server. Điều này cho phép bạn xây dựng các ứng dụng IoT, quản lý dữ liệu và triển khai các luồng làm việc tự động trên môi trường máy chủ.
* **Đám mây:** Node-RED có thể được triển khai trên các nền tảng đám mây như Amazon Web Services (AWS), Microsoft Azure hoặc Google Cloud Platform (GCP). Các nhà cung cấp đám mây này thường cung cấp dịch vụ quản lý và triển khai Node-RED dưới dạng các dịch vụ dựa trên đám mây, giúp bạn xây dựng và chạy các luồng làm việc trên cơ sở hạ tầng đám mây.
* **Thiết bị IoT:**Node-RED có thể được triển khai trực tiếp trên các thiết bị nhúng như Raspberry Pi, Arduino hoặc các thiết bị IoT khác. Điều này cho phép bạn xây dựng các ứng dụng IoT phức tạp trên các thiết bị có tài nguyên hạn chế và tương tác với các cảm biến, thiết bị và dịch vụ khác trong mạng IoT.

**Node-RED** là một công cụ linh hoạt và có thể được triển khai trên nhiều môi trường khác nhau tùy thuộc vào nhu cầu và yêu cầu của dự án.

* 1. **Ứng dụng của Node-RED.**

Node-RED có rất nhiều ứng dụng trong các lĩnh vực khác nhau. Dưới đây là một số ví dụ về ứng dụng của Node-RED:

* **Internet of Things (IoT):** Node-RED là một công cụ mạnh mẽ để xây dựng các ứng dụng IoT. Bạn có thể sử dụng Node-RED để kết nối, thu thập dữ liệu và điều khiển các thiết bị IoT như cảm biến, bộ điều khiển và thiết bị mạng. Bạn cũng có thể tạo các luồng làm việc để xử lý dữ liệu, giám sát và phản ứng đối với sự kiện từ các thiết bị IoT.
* **Tự động hóa quy trình:** Node-RED cho phép bạn tự động hóa các quy trình và công việc. Bạn có thể xây dựng các luồng làm việc để thực hiện các nhiệm vụ như quản lý lịch trình, gửi thông báo, xử lý dữ liệu, tương tác với hệ thống và dịch vụ khác. Node-RED giúp giảm thiểu công sức và thời gian làm việc cần thiết cho các quy trình tự động.
* **Xử lý dữ liệu và tích hợp hệ thống:** Node-RED cung cấp các công cụ và nút để xử lý dữ liệu, chuyển đổi định dạng, tính toán và lưu trữ dữ liệu. Bạn có thể sử dụng Node-RED để tích hợp và tương tác với các hệ thống và dịch vụ khác nhau như cơ sở dữ liệu, hệ thống tệp, giao thức web, API và dịch vụ đám mây.
* **Giám sát và quản lý:**Node-RED cho phép bạn xây dựng các bảng điều khiển (dashboard) và giao diện người dùng để giám sát và quản lý các quy trình, dữ liệu và thiết bị. Bạn có thể tạo các trang web tương tác, biểu đồ, bảng và các yếu tố trực quan khác để hiển thị thông tin và điều khiển hoạt động của hệ thống.
* **Phân tích dữ liệu và trí tuệ nhân tạo:**Node-RED cung cấp tích hợp với các công cụ và thư viện phân tích dữ liệu và trí tuệ nhân tạo như TensorFlow và Node.js Machine Learning. Bạn có thể sử dụng Node-RED để xây dựng các mô hình, phân loại dữ liệu, nhận dạng hình ảnh.
  1. **Cài đặt Node-Red trên Windowns.**

Node-RED có rất nhiều ứng dụng trong các lĩnh vực khác nhau. Dưới đây là một số ví dụ về ứng dụng của Node-RED:

* **Internet of Things (IoT):** Node-RED là một công cụ mạnh mẽ để xây dựng các ứng dụng IoT. Bạn có thể sử dụng Node-RED để kết nối, thu thập dữ liệu và điều khiển các thiết bị IoT như cảm biến, bộ điều khiển và thiết bị mạng. Bạn cũng có thể tạo các luồng làm việc để xử lý dữ liệu, giám sát và phản ứng đối với sự kiện từ các thiết bị IoT.
* **Tự động hóa quy trình:** Node-RED cho phép bạn tự động hóa các quy trình và công việc. Bạn có thể xây dựng các luồng làm việc để thực hiện các nhiệm vụ như quản lý lịch trình, gửi thông báo, xử lý dữ liệu, tương tác với hệ thống và dịch vụ khác. Node-RED giúp giảm thiểu công sức và thời gian làm việc cần thiết cho các quy trình tự động.
* **Xử lý dữ liệu và tích hợp hệ thống:** Node-RED cung cấp các công cụ và nút để xử lý dữ liệu, chuyển đổi định dạng, tính toán và lưu trữ dữ liệu. Bạn có thể sử dụng Node-RED để tích hợp và tương tác với các hệ thống và dịch vụ khác nhau như cơ sở dữ liệu, hệ thống tệp, giao thức web, API và dịch vụ đám mây.
* **Giám sát và quản lý:**Node-RED cho phép bạn xây dựng các bảng điều khiển (dashboard) và giao diện người dùng để giám sát và quản lý các quy trình, dữ liệu và thiết bị. Bạn có thể tạo các trang web tương tác, biểu đồ, bảng và các yếu tố trực quan khác để hiển thị thông tin và điều khiển hoạt động của hệ thống.
* **Phân tích dữ liệu và trí tuệ nhân tạo:**Node-RED cung cấp tích hợp với các công cụ và thư viện phân tích dữ liệu và trí tuệ nhân tạo như TensorFlow và Node.js Machine Learning. Bạn có thể sử dụng Node-RED để xây dựng các mô hình, phân loại dữ liệu, nhận dạng hình ảnh.

1. **NodeJS**
   1. **NodeJS là gì?**

* Node.js là một mã nguồn mở, một môi trường cho các máy chủ và ứng dụng mạng.
* Node.js sử dụng Google V8 JavaScript engine để thực thi mã, và một tỷ lệ lớn các mô-đun cơ bản được viết bằng JavaScript. Các ứng dụng node.js thì được viết bằn JavaScript.
* Node.js chứa một thư viện built-in cho phép các ứng dụng hoạt động như một Webserver mà không cần phần mềm như Nginx, Apache HTTP Server hoặc IIS.
* Node.js cung cấp kiến trúc hướng sự kiện (event-driven) và non-blocking I/O API, tối ưu hóa thông lượng của ứng dụng và có khả năng mở rộng cao
* Mọi hàm trong Node.js là không đồng bộ (asynchronous). Do đó, các tác vụ đều được xử lý và thực thi ở chế độ nền (background processing)

### Ứng dụng của NodeJS

* Xây dựng websocket server (Chat server)
* Hệ thống Notification (Giống như facebook hayTwitter)
* Ứng dụng upload file trên client
* Các máy chủ quảng cáo
* Các ứng dụng dữ liệu thời gian thực khác.
  1. **Ưu nhược điểm của NodeJS**
     1. **Nhược điểm NodeJS**

Ứng dụng nặng tốn tài nguyên Nếu bạn cần xử lý các ứng dụng tốn tài nguyên CPU như encoding video, convert file, decoding encryption… hoặc các ứng dụng tương tự như vậy thì không nên dùng NodeJS (Lý do: NodeJS được viết bằng C++ & Javascript, nên phải thông qua thêm 1 trình biên dịch của NodeJS sẽ lâu hơn 1 chút ). Trường hợp này bạn hãy viết 1 Addon C++ để tích hợp với NodeJS để tăng hiệu suất tối đa !

NodeJS và ngôn ngữ khác NodeJS, PHP, Ruby, Python .NET …thì việc cuối cùng là phát triển các App Web. NodeJS mới sơ khai như các ngôn ngữ lập trình khác. Vậy nên bạn đừng hi vọng NodeJS sẽ không hơn PHP,Ruby,Python… ở thời điểm này. Nhưng với NodeJS bạn có thể có 1 ứng dụng như mong đợi, điều đó là chắc chắn !

* + 1. **Ưu điểm NodeJS**

Đặc điểm nổi bật của Node.js là nó nhận và xử lý nhiều kết nối chỉ với một single-thread. Điều này giúp hệ thống tốn ít RAM nhất và chạy nhanh nhất khi không phải tạo thread mới cho mỗi truy vấn giống PHP. Ngoài ra, tận dụng ưu điểm non-blocking I/O của Javascript mà Node.js tận dụng tối đa tài nguyên của server mà không tạo ra độ trễ như PHP

JSON APIs Với cơ chế event-driven, non-blocking I/O(Input/Output) và mô hình kết hợp với Javascript là sự lựa chọn tuyệt vời cho các dịch vụ Webs làm bằng JSON.

Ứng dụng trên 1 trang( Single page Application) Nếu bạn định viết 1 ứng dụng thể hiện trên 1 trang (Gmail?) NodeJS rất phù hợp để làm. Với khả năng xử lý nhiều Request/s đồng thời thời gian phản hồi nhanh. Các ứng dụng bạn định viết không muốn nó tải lại trang, gồm rất nhiều request từ người dùng cần sự hoạt động nhanh để thể hiện sự chuyên nghiệp thì NodeJS sẽ là sự lựa chọn của bạn.

Shelling tools unix NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra 1 luồng khiến cho hiệu xuất hoạt động đạt mức tối đa nhất và tuyệt vời nhất.

Streamming Data (Luồng dữ liệu) Các web thông thường gửi HTTP request và nhận phản hồi lại (Luồng dữ liệu). Giả xử sẽ cần xử lý 1 luồng giữ liệu cực lớn, NodeJS sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.

Ứng dụng Web thời gian thực Với sự ra đời của các ứng dụng di động & HTML 5 nên Node.js rất hiệu quả khi xây dựng những ứng dụng thời gian thực (real-time applications) như ứng dụng chat, các dịch vụ mạng xã hội như Facebook, Twitter,…

* + 1. **Với những ưu - nhược trên, khi nào chúng ta sẽ dùng Node.js?**

Node.js rất hấp dẫn. Nhưng khi bạn quyết định bắt tay xây dựng một dự án bằng Node.js, hãy đặt câu hỏi: “Tôi có nên dùng Node.js hay không?”. Và dưới đây là một trong số những câu trả lời cho điều đó.

**Bạn KHÔNG nên sử dụng Node.js khi:**

* Xây dựng các ứng dụng hao tốn tài nguyên: Bạn đừng mơ mộng đến Node.js khi bạn đang muốn viết một chương trình convert video. Node.js hay bị rơi vào trường hợp thắt cổ chai khi làm việc với những file dung lượng lớn.
* Một ứng dụng chỉ toàn CRUD: Node.js không nhanh hơn PHP khi bạn làm các tác vụ mang nặng tính I/O như vậy. Ngoài ra, với sự ổn định lâu dài của các webserver script khác, các tác vụ CRUD của nó đã được tối ưu hóa. Còn Node.js? Nó sẽ lòi ra những API cực cực kỳ ngớ ngẩn.
* Khi bạn cần sự ổn định trong ứng dụng của bạn: Chỉ với 4 năm phát triển của mình (2009-2013), version của Node.js đã là 0.10.15 (hiện tại tới thời điểm này là v0.10.35). Mọi API đều có thể thay đổi – một cách không tương thích ngược – hãy thật cẩn thận với những API mà bạn đang dùng, và luôn đặt câu hỏi: “Khi nó thay đổi, nó sẽ ảnh hưởng gì đến dự án của tôi?”
* Và quan trọng nhất: Bạn chưa hiểu hết về Node.js Node.js cực kỳ nguy hiểm trong trường hợp này, bạn sẽ rơi vào một thế giới đầy rẫy cạm bẫy, khó khăn. Với phần lớn các API hoạt động theo phương thức non-blocking/async việc không hiểu rõ vấn đề sẽ làm cho việc xuất hiện những error mà thậm chí bạn không biết nó xuất phát từ đâu? Và mệt mỏi hơn nữa: Khi cộng đồng Node.js chưa đủ lớn mạnh, và sẽ ít có sự support từ cộng đồng. Khi mà phần lớn cộng đồng cũng không khá hơn bạn là bao.

**Vậy bạn nên dùng Node.js khi nào?**

* Node.js thực sự tỏa sáng trong việc xây dựng RESTful API (json). Gần như không có ngôn ngữ nào xử lý JSON dễ dàng hơn Javascript, chưa kể các API server thường không phải thực hiện những xử lý nặng nề nhưng lượng concurrent request thì rất cao. Mà Node.js thì xử lý non-blocking. Chẳng còn gì thích hợp hơn Node.js trong trường hợp này!
* Những ứng dụng đòi hỏi các giao thức kết nối khác chứ không phải chỉ có http. Với việc hỗ trợ giao thức tcp, từ nó bạn có thể xây dựng bất kỳ một giao thức custom nào đó một cách dễ dàng.
* Những ứng dụng thời gian thực: Khỏi phải nói vì Node.js dường như sinh ra để làm việc này!
* Những website stateful. Node.js xử lý mọi request trên cùng một process giúp cho việc xây dựng các bộ nhớ đệm chưa bao giờ đơn giản đến thế: Hãy lưu nó vào một biến global, và thế là mọi request đều có thể truy cập đến bộ nhớ đệm đó. Caching sẽ không còn quá đau đầu như trước đây, và bạn có thể lưu cũng như chia sẻ trạng thái của một client với các client khác ngay trong ngôn ngữ, chứ bạn không cần thông qua các bộ nhớ ngoài!
* Quan trọng nhất: Bạn yêu thích và muốn sử dụng nó.

1. **Triển khai, trải nghiệm và ứng dụng của phần mềm.**
2. **Dữ liệu OpenEEW.**
   1. **Truy cập dữ liệu OpenEEW trên AWS.**

Các tệp được sắp xếp thành các thư mục dựa trên quốc gia, thiết bị và ngày UTC tương ứng với dữ liệu; logic chính xác để gán dữ liệu cho một tệp dựa trên ngày được giải thích chi tiết hơn bên dưới. Đường dẫn tệp thông thường sẽ có dạng:

records/

country\_code=<country\_code>/

device\_id=<device\_id>/

year=<year>/

month=<month>/

day=<day>/

hour=<hour>/

<minute>.jsonl

Chú thích:

• <country\_code>: là mã quốc gia gồm hai chữ số ISO 3166 của quốc gia nơi thiết bị được triển khai, dưới dạng chữ thường, ví dụ: mx đối với Mexico, cl đối với Chile.

• <device\_id>: có độ dài thay đổi, là mã định danh chuỗi của thiết bị, ví dụ: 000, 001.

• Mã này có thể được sử dụng cùng với tệp siêu dữ liệu của thiết bị (xem bên dưới). Nó là duy nhất cho mỗi quốc gia

• <year>: là năm UTC với thế kỷ của dữ liệu, ví dụ: 2019

• <month>: là tháng UTC của dữ liệu dưới dạng số thập phân không đệm, ví dụ: 01, 02, 11, 12

• <day>: là ngày UTC của dữ liệu dưới dạng số thập phân không đệm, ví dụ: 01, 31

• <hour>: là giờ UTC (đồng hồ 24 giờ) của dữ liệu dưới dạng số thập phân không đệm, ví dụ 00, 23

• <minute>: là phút UTC được sử dụng để gắn nhãn cho nhóm 5 phút chứa dữ liệu dưới dạng số thập phân không đệm, ví dụ: 00, 05, 55

• Phần mở rộng tệp .jsonl cho biết dữ liệu được lưu trữ dưới dạng đối tượng JSON được phân tách bằng dòng mới.

* 1. **Trường dữ liệu.**

Dữ liệu gia tốc kế được lưu trữ với cấu trúc giống như nhận được từ thiết bị để mọi quá trình xử lý áp dụng cho dữ liệu đã lưu trữ đều có thể được áp dụng trực tiếp vào dữ liệu theo thời gian thực.

Mỗi hàng tệp chứa một đối tượng JSON với các trường sau:

country\_code: giống như tham số thư mục.

device\_id: giống như tham số thư mục.

x: mảng các giá trị thập phân trục x của gia tốc kế tính bằng gals (cm/s2).

y: mảng các giá trị thập phân trục y của gia tốc kế tính bằng gals (cm/s2).

z: mảng các giá trị thập phân trục z của gia tốc kế tính bằng gals (cm/s2).

device\_t: thời gian Unix của thiết bị tương ứng với giá trị cuối cùng của x, y và z mảng.

cloud\_t: Thời gian Unix khi dữ liệu được nhận trên đám mây.

sr: tốc độ mẫu, số lượng giá trị trên giây.

Lưu ý rằng các mảng và x có cùng độ dài và tương ứng với cùng thời gian đo, tức là giá trị thứ n trong các mảng và tương ứng với cùng thời gian đo. Độ dài của mảng, thường là 32 mỗi giây, không phụ thuộc vào tốc độ mẫu .yzxyzsr

Thông thường cloud\_t và device\_t chênh lệch ít hơn một giây, nhưng đôi khi thời gian của thiết bị bị lệch nên chúng tôi khuyên bạn nên sử dụng cloud\_t làm dấu thời gian đo đáng tin cậy hơn. Việc sắp xếp dữ liệu trước cloud\_t và sau đó device\_tđược đề xuất trong những trường hợp hiếm hoi khi hai bản ghi có cùng giá trị cloud\_t.

Lưu ý rằng thứ tự dữ liệu trong một tệp nhất định không được đảm bảo.

Dấu thời gian riêng lẻ có thể được gán cho từng điểm dữ liệu trong mảng bằng cách sử dụng các giá trị cloud\_t (hoặc device\_t) và sr, tức là bằng cách trừ dấu thời gian bội số 1/sr dựa trên giá trị chỉ mục mảng.

* 1. **Cách các bản ghi được gán cho một tập tin.**

Các bản ghi được gán cho một tệp dựa trên giá trị UTC của cloud\_t trường.Phút, MM, của tệp, cùng với năm, tháng, ngày và giờ chứa trong đó phải sao cho:

year-month-day hour:MM <= cloud\_t < year-month-day hour:MM + 5 minutes

* 1. **Siêu dữ liệu thiết bị**

Siêu dữ liệu của thiết bị được lưu trữ trong đường dẫn tệp sau:

devices/

country\_code=<country\_code>/

devices.jsonl

Mỗi JSON chứa các trường sau:

country\_code: giống như tham số thư mục bản ghi.

device\_id: giống như tham số thư mục bản ghi.

latitude: vĩ độ thiết bị đến 2 chữ số thập phân.

longitude: kinh độ thiết bị đến 2 chữ số thập phân.

effective\_from: số nguyên thời gian Unix mà siêu dữ liệu hợp lệ.

effective\_to: số nguyên thời gian Unix cho đến khi siêu dữ liệu hợp lệ.

is\_current\_row: hoặc true để false cho biết liệu hàng hiện có hiệu lực hay không, thay thế cho việc kiểm tra ngày có hiệu lực khi tìm kiếm giá trị hiện tại.

vertical\_axis: 'x', 'y' hoặc 'z' để chỉ định trục nào có hướng thẳng đứng.

horizontal\_axes: danh sách hai trục có hướng nằm ngang.

Và sao cho không có sự trùng lặp, nên chỉ cần kiểm tra là:

effective\_from <= Unix time <= effective\_to

1. **Khai triển ứng dụng**

Một cách tiếp cận đơn giản để phát hiện hoạt động địa chấn là đo gia tốc lớn từ các cảm biến gia tốc kế openeew, dựa trên ngưỡng đã đặt cao hơn mức nhiễu điển hình của gia tốc kế. Wikipedia định nghĩa gal (đơn vị) là đơn vị gia tốc được sử dụng rộng rãi trong khoa học đo trọng lực. Gal được định nghĩa là bình phương 1 cm trên giây (1 cm/s2). Hoạt động địa chấn đáng quan tâm có thể vượt quá 3 gals (3 cm/giây2).

Các luồng Node-RED này quan sát dữ liệu gia tốc kế openeew theo thời gian thực và tính toán xem cảm biến có thể gặp phải hoạt động địa chấn hay không bằng thuật toán sau. Mỗi giây, hàm này nhận được mảng x/y/z dữ liệu rung động dưới giây. Các mảng dữ liệu được truyền vào hàm bên trong msg.payload.traces[0] Hàm javascript lặp qua dữ liệu rung để tìm kiếm gia tốc vượt quá 3 cm/giây2Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* Luồng này hiển thị bảng điều khiển Node-RED hiển thị cho nhà điều tra / nhà địa chấn học lịch, các tùy chọn khoảng thời gian và danh sách các cảm biến OpenEEW. Người điều tra chọn một cảm biến thú vị và khoảng thời gian để nghiên cứu và truy vấn bộ dữ liệu OpenEEW. Sau đó, luồng sẽ vẽ dữ liệu cảm biến lịch sử trong một tập hợp biểu đồ.
* Luồng xây dựng một URL cho cảm biến và khoảng thời gian đã chọn, đồng thời truy xuất dữ liệu cảm biến lịch sử từ tập dữ liệu OpenEEW và phát lại thành biểu đồ trong Bảng điều khiển Node-RED. Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

  Mô tả được tạo tự động
  + Cách thành phần gồm:
    - Chọn ngày
    - Chọn giờ
    - Chọn khoảng thời gian
  + Kết quả hiển thị là Ảnh có chứa văn bản, hàng, Phông chữ, số

    Mô tả được tạo tự động
* Chọn cả biến để vẽ biểu đồ: có tác dụng hiển thị các thiết bị cảm biến ở Mexico và có thể chọn để nhận dữ liệu từ <https://s3.amazonaws.com/grillo-openeew/index.html> và vẽ biểu đồ để hiển thị trạng thái
  + Các thành phần gồm :
    - UI control : để có thể thao tác trên user interface sau khi được tạo ra
    - Sensor Location: chứa dữ liệu từ vị trí của các thiết bị cảm biến ( device\_id, latitude, longitude, location) Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

      Mô tả được tạo tự động
    - CSV: để đọc dữ liệu từ node Sensor Location sau đó render ra bảng

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

* + Kết quả hiển thị là Ảnh có chứa văn bản, ảnh chụp màn hình, số, Song song

    Mô tả được tạo tự động
* Bulid and hiển thị OpenEEW dataset để nhận dữ liệu thì file nguồn Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

  Mô tả được tạo tự động
  + Các thành phần gồm :
    - Build filename

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

* + - Code khởi tạo các biến ts, hour, minute, và device từ các giá trị trong ngữ cảnh của luồng Node-RED. Nếu những giá trị này không có trong luồng, giá trị mặc định như "[missing]" sẽ được gán.
    - year, month, và day được khởi tạo với giá trị mặc định là "[missing]".
    - Nếu ts (thời gian dấu thời) có giá trị, code sẽ tạo ra một đối tượng ngày từ ts và rút trích năm (year), tháng (month), và ngày (day) từ đối tượng này.
    - Dựa trên giá trị của device, tên tệp sẽ được xây dựng khác nhau. Nếu device có giá trị là '3ef3d787af85' hoặc 'db0f06aa9270', tên tệp sẽ chứa thông tin về country\_code=pr. Ngược lại, nó sẽ chứa thông tin về country\_code=mx.
  + Build a html link to the dataset: để tạo ra link html và chỉnh sữa font chữ

Ảnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

* + Kết quả hiển thị



* Nút "next" trong bảng điều khiển Node-RED Phát lại trận động đất lịch sử cực kỳ sâu sắc. Trọng tâm thường là các trận động đất lớn và dấu thời gian/bộ dữ liệu cụ thể của chúng. Nút "next" cho phép điều tra viên khám phá vô số dư chấn. Bắt đầu với trận động đất mạnh 7,2 độ richter ở Mexico vào lúc 23:35 ngày 16 tháng 2 năm 2018 và tập dữ liệu của nó, sau đó nhấp vào nút Tiếp theo để xem (với khoảng tăng 5 phút) các dư chấn xảy ra trong vài giờ tiếp theo. Xem ảnh gif hoạt hình về trận động đất và hoạt động dư chấn. Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

  Mô tả được tạo tự động
  + Vì cứ mỗi 5 phút thì dữ liệu sẽ cập nhật 1 lần nên “next” botton sẽ cài đặt là cộng 5 phút sau khi nhấn

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

* + Sau khi có được thời gian sau 5 phút , ta chỉnh lại thời gian đã hiển thị ở phần trên bằng node “Set filename” và cho nó trỏ đến phần build filename Ảnh có chứa ảnh chụp màn hình, biểu đồ, hàng, văn bản

    Mô tả được tạo tự độngẢnh có chứa văn bản, ảnh chụp màn hình, hàng, biểu đồ

    Mô tả được tạo tự động
* Download a historical sensor dataset: tải xuống file JSON trong dataset. Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

  Mô tả được tạo tự động
  + Có các thành phần như :
    - Botton “Retrieve data and plot” : nhấn vào thì sẽ lấy dữ liệu từ dataset và vẽ biểu đồ
    - Switch gồm 2 nhanh là missing and none missing
      * Nếu missing thì sẽ hiển thị message error ra màn hình



* + - * Ngược lại thì sẽ set filename ra đường dẫn của dataset và get data về
* Plot the seismic graphs: render thông tin cảm biến ra màn hình dưới dạng biểu đồ Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

  Mô tả được tạo tự động
  + Có các thành phần gồm :
    - Nút “Reset” : để trả về biểu đồ trống và vẽ lại từ đầu Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

      Mô tả được tạo tự động
    - Build Quake Charts: thông qua dữ liệu đã get về từ “get OpenEEW data” và vẽ biểu đồ

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

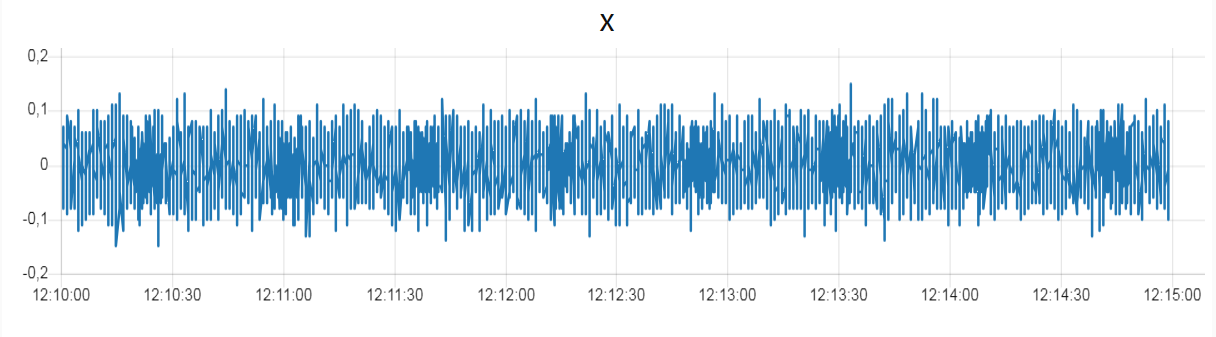
Mô tả được tạo tự động

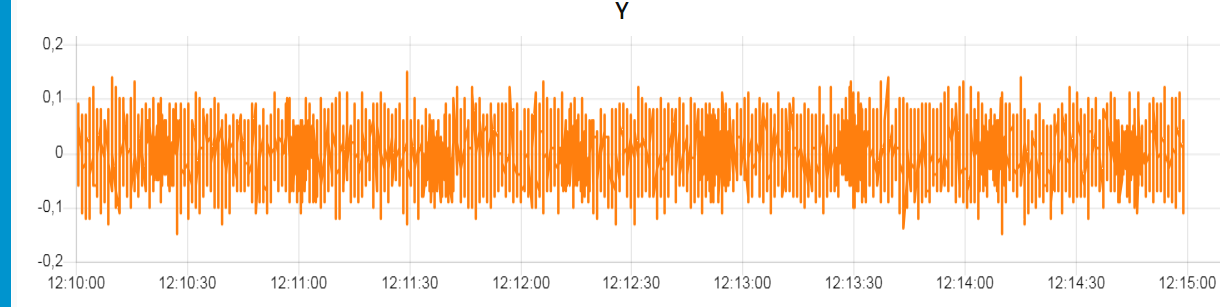
* + - Sau đây là code JavaScript thông qua việc đọc dữ liệu từ file JSON và vẽ lênh Quake Charts

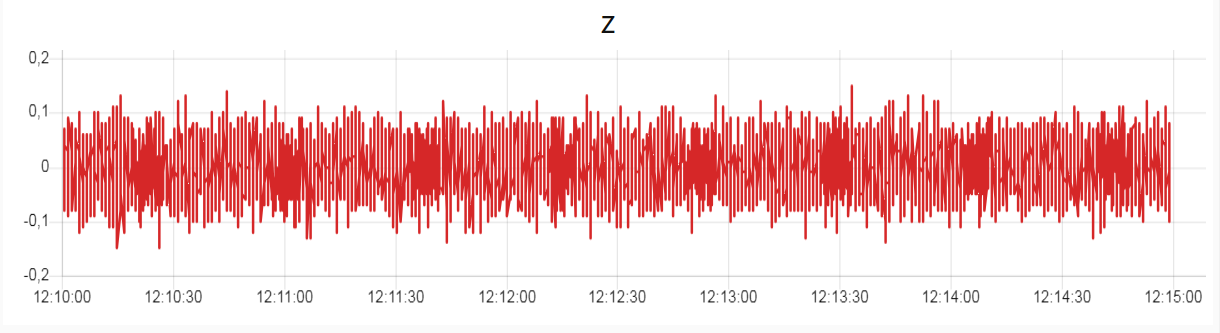
Ảnh có chứa văn bản, ảnh chụp màn hình, tài liệu, Phông chữ

Mô tả được tạo tự động

**Ta thu được kết quả hiển thị:**







1. **Kết luận và hướng phát triển.**

Hiện nay, có nhiều người dân sống ở những nơi thường xuyên xảy hiện tượng động đất và sóng thần. Mỗi năm có rất nhiều người dân đã phải thiệt mạng và rất nhiều tài sản đã mất đi trong những đợt động đất, sóng thần. Các chính quyền đã có rất nhiều các biện pháp để bảo vệ người dân. Trong đó biện pháp chuẩn đoán và cảnh báo để người dân di tản đến nơi an toàn là các tốt nhất.

Chúng em tạo ra phần mềm này để có thể cảnh báo cho người dân ở những nơi thường xuyên xảy ra động đất về mức độ rung chấn ở nơi họ đang sống một cách nhanh chóng. Để có khi có rung chấn mạnh đến mức báo động thì người dân có thể di tản tới những vùng an toàn để có thể bảo vệ được tính mạng và giảm thiếu những thiệt hại của động đất tới con người cũng như những tài sản của người dân.

Chúng em mong muốn ứng dụng sẽ nâng cấp thêm các tính năng và được phát triển rộng rãi trên khắp thế giới, để những người dân ở các nơi thường xuyên xảy ra động đất có thể bảo vệ được tài sản và tính mạng của bản thân mình.

**Tư liệu tham khảo**

[1] Grillo - Initial work, link: [Grillo](https://grillo.io/)

[2] Grillo - OpenEEW Explorer, link : <https://s3.amazonaws.com/grillo-openeew/index.html>

[3] Ngọc OUTOMATION (29/12/2023), link: <https://ngocautomation.com/node-red-la-gi/>

[4] Link: <https://github.com/openeew/openeew/tree/master/data?fbclid=IwAR2clgBUv-PIg3hQcEOG9UwOOcX7jsQtwOiw35iy0fqdTL1QqT2u8NF71ak#accessing-openeew-data-on-aws>

[5] Openeew, link: <https://github.com/openeew/openeew-nodered?tab=readme-ov-file&fbclid=IwAR0EO9qZ0zKcRpyGNhCtumIZaUHO8EGMSysiqAEryr_pPXe7z8xihOG6n_c>