



Departamento de Computación e Informática
Computación Paralela y Distribuida

PROYECTO REST

Profesor: Sebastián Salazar Molina

Integrantes: Alonso Pino Chung - Valentina Tobar Alarcón

Fecha entrega: 30 Octubre 2021



ÍNDICE

INTRODUCCIÓN	3
FORMA DE RESOLVER EL PROBLEMA	4
TECNOLOGIA UTILIZADA.....	6
CONCLUSIÓN	8

INTRODUCCIÓN

En un mundo que avanza a pasos agigantados en la gestión y desarrollo de la informática, se proponen nuevos desafíos para situaciones que han significado problemáticas a nivel mundial durante largos años. Es por esto, que la realización de este proyecto nos permite acercarnos a la búsqueda de soluciones para estos planteamientos, aplicando nuestro conocimiento y aplicando conceptos aprendidos en la asignatura.

En específico trabajaremos información referente a las precipitaciones en nuestro país, un tema contingente actualmente, cuando a nivel nacional atravesamos una de las mayores crisis hídricas de la última década. La disminución de las precipitaciones trae consigo consecuencias dañinas para el medio ambiente, por lo que la recopilación de esta información será de ayuda para el estudio y comparación de datos con respecto a años anteriores.

La información histórica de las precipitaciones se encuentra disponible en distintos sitios web, por lo que mediante un API REST recopilaremos los distintos datos, permitiendo en un futuro consumirla y poder realizar consultas con respecto a precipitaciones en distintas regiones.

Un API es un conjunto de definiciones y protocolos que tienen la finalidad de servir como herramienta para desarrollar e integrar el software de las aplicaciones, de esta forma uno de los principales beneficios es el ahorro de tiempo al momento de trabajar con información. Un API REST corresponde a un conjunto de límites de arquitectura, uno de los beneficios es la comunicación al momento de transportar datos, ya que utiliza el protocolo HTTP y nos permite utilizar diversos métodos como GET, POST, PUT entre otros.

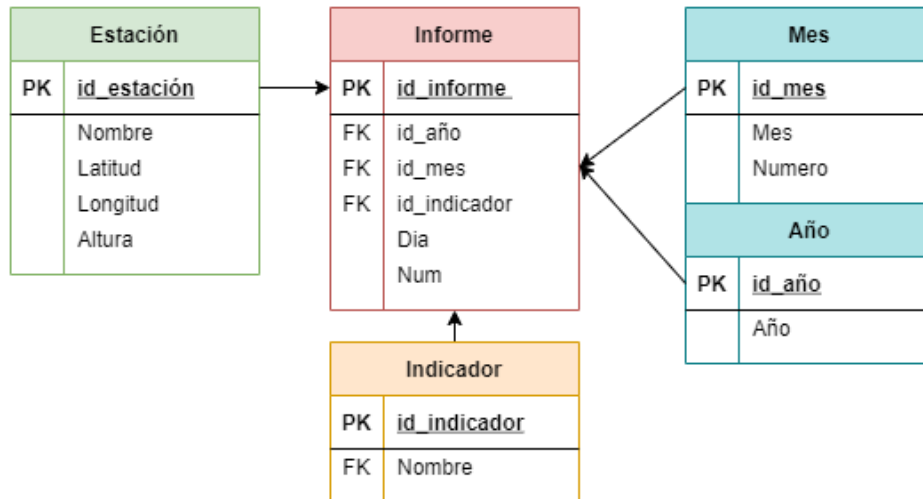
FORMA DE RESOLVER EL PROBLEMA

Se requiere la recopilación de información histórica sobre las precipitaciones a nivel nacional desde el año 1950, obteniendo dichos datos de tres sitios web distintos, esta información corresponde a estudios realizados por la Universidad de Chile y el gobierno de Chile.

La mejor forma de resolver este problema es mediante el **Web Scrapping**, esta corresponde a una técnica que nos permite extraer la información de páginas web de manera automatizada en tiempo real, de este modo podemos extraer los datos a diario de las precipitaciones en las estaciones y trabajarlos a nuestra conveniencia.

Para esto se realizará la creación de un servidor local en un proyecto Node.js, esto será posible con la implementación del módulo de desarrollo Express. Al realizar la configuración del servidor local, le designamos el puerto 3000, y aquí es donde realizaremos las pruebas posteriormente en Postman.

Para el desarrollo del web scrapping utilizaremos el módulo cheerios, que nos facilita la obtención de datos de desde un sitio. Este módulo obtiene la dirección .html del sitio web y mediante funciones podemos obtener datos específicos de él, en este caso la tabla de datos, posteriormente estos son enviados a la base de datos PostgreSQL previamente creada con sus respectivas tablas y relaciones.



Modelo Físico Relacional

Con los archivos Excel se utilizará el módulo XLSX, esta herramienta nos permite obtener los datos de hojas de cálculo para su posterior manipulación, en nuestro caso obtuvimos las hojas de cálculo .CSV desde el sitio web del Departamento de Geofísica de la Universidad de Chile. De la misma manera que con el primer sitio web, almacenaremos los datos en la base de datos.

Las funciones utilizadas para responder a este tipo de peticiones son asíncronas y se realizaron mediante “Async y Await”, de esta manera las respuestas a las peticiones realizadas a la base de datos son precisas y claras, permitiendo entregar el respectivo status code, los datos retornados o el respectivo error detectado.

Una vez la aplicación logra recopilar los datos de estos sitios web, procedemos a realizar la programación que permiten gestionar un usuario con la ayuda de los módulos DotEnv, Bcryptjs y Jsonwebtoken; estos principalmente están enfocados a la seguridad del usuario.

Realizar la gestión de usuarios es totalmente necesario ya que para realizar la consulta de datos el usuario debe estar previamente registrado. Luego de gestionados los usuarios, la obtención y almacenamiento de datos; estamos en condiciones de realizar peticiones mediante PostMan.

Ahora los datos están listos y almacenados, para ser posteriormente consumidos en alguna aplicación web o móvil.

TECNOLOGIA UTILIZADA

El desarrollo de este proyecto se realizó casi en su totalidad en Windows, aunque durante el inicio de la programación utilizamos Ubuntu mediante una máquina virtual. En ambos sistemas operativos utilizamos la herramienta de desarrollo de código **Visual Studio Code**, debido a su simpleza y gran variedad de extensiones que facilitan la realización del proyecto. Además, para un trabajo continuo y a la par entre compañeros de proyecto utilizamos **GitHub**, herramienta que nos facilita el manejo y gestión de control de versiones del código.

Para el desarrollo del código evaluamos las diversas opciones que nos permitían crear un API REST, para esto tomamos en consideración el conocimiento de cada uno de los integrantes, la curva de aprendizaje y disponibilidad nuestra para el desarrollo del código.

Luego de analizar la situación, consideramos que la que más se acomodó a nuestra idea de trabajo fue el entorno de ejecución de JavaScript llamado **NodeJs**, dentro de los beneficios que ofrece este entorno encontramos la posibilidad de utilizar distintos tipos de módulos vía **NPM** (Node Package Manager) que nos facilitan el trabajo a la hora de programar. A continuación, se especificarán los módulos más importantes que utilizamos a lo largo del desarrollo del proyecto:

- **Nodemon**: Es una utilidad que monitorea los cambios en el código fuente que se está desarrollando y automáticamente re inicia el servidor.
- **Cheerio**: Modulo que analiza un sitio en específico y proporciona una API para atravesar / manipular la estructura de datos resultante.
- **Cors**: Es un mecanismo para permitir o restringir los recursos solicitados en un servidor web dependiendo de dónde se inició la solicitud HTTP
- **DotEnv**: Nos permite leer variables desde un archivo llamado .env, para posteriormente cargar las variables de entorno en la variable process.env.
- **Bcryptjs** : Una biblioteca para ayudarte a codificar contraseñas.
- **Jsonwebtoken**: Es una herramienta que nos permite autenticarnos con el servidor mediante Tokens de una forma simple y segura.
- **Express**: Herramienta que proporciona un conjunto sólido de funciones para las conexiones en aplicaciones web y móviles.
- **Pg**: Modulo que facilita la interacción con la base de datos PostgreSQL.
- **Request**: Está diseñada para ser la forma más sencilla posible de realizar llamadas http. Es compatible con HTTPS y sigue las redirecciones de forma predeterminada.
- **Babel**: Herramienta que nos permite transformar nuestro código JS de última generación a un código de JavaScript que cualquier navegador o versión de Node. js pueda entender.
- **XLSX**: Es un analizador y escritor para varios formatos de hojas de cálculo.

Para la gestión de base de datos, utilizamos **PostgreSQL**, debido a que es una herramienta de código abierto, fácil de manipular y nos ha dado buenos resultados en trabajos anteriores.

Además, se implementó **Docker** de esta manera se empaqueta el software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que este se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución.

Para realizar las consultas al servidor utilizamos **PostMan**, que es un software que nos permite realizar pruebas en API REST.

CONCLUSIÓN

La realización de este trabajo nos permitió comprender y aplicar de mejor manera el concepto de Web Scrapping, en el proyecto de creación de una API REST. De esta manera comenzamos a dimensionar la importancia de la gestión y almacenamiento de datos tanto como herramienta de estudio, prevención e información. Esta nos será de ayuda para algún proyecto posterior en el que podamos poner a disposición de la sociedad este tipo de contenido y pueda ser utilizado para su estudio o la toma de conciencia del cambio climático que nos asecha actualmente.

Además, comprendimos que la utilización de un API REST nos entrega independencia al momento de tener un consumidor, permitiendo el consumo de esta de diversas formas y para diversos usos, es por estas ventajas que la convierte en una de las arquitecturas web más utilizadas