



Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática



Proyecto: Mapler

Entrega parcial 3








Por Sebastián García Delgadillo

Asignatura: INF3242 - Ingeniería Web
Profesora: Sandra Cano Mazuera

Análisis de backend

Para mantener los datos relacionados a las aplicaciones web de **Mapler**, es necesario el desarrollo de un backend robusto y confiable, utilizando las herramientas disponibles para ello. Para el desarrollo del backend, se ha decidido por la creación de un servicio web de API HTTP¹, utilizando peticiones HTTP y lógica de tokens para seguridad.

Las decisiones respecto a las tecnologías y frameworks a utilizar en el desarrollo de la API, se resume en lo siguiente:

- Framework** : Para la API HTTP, por conocimiento previo del framework, y por simplicidad a la hora de programar, dado que se utilizará el mismo lenguaje tanto en frontend como en backend, se decidió por el uso del framework  Express, y en su desarrollo se seguirá el patrón de diseño **modelo-vista-controlador (MVC)**.
- Lenguaje** : Como lenguaje de programación para la aplicación de express, se eligió  Typescript, ya que los tipos permiten minimizar errores de lógica.
- Base de datos** : Debido a la estructura del sitio, y que React es una librería orientada a documentos, se decidió por almacenar los datos en una base de datos No-SQL. Por simplicidad, y por conocimiento previo de mongoose, se decidió por usar  MongoDB
- Utilidades/Despliegue** : Para el despliegue de la API, se decidió el uso de soluciones *sin servidor*. Para ello, se hará uso de la prueba gratuita ofrecida por  Amazon Web Services, y así utilizar  API Gateway y  Lambda para el despliegue de la aplicación de express, en conjunto con  Serverless para facilitar el proceso.

La utilización de **Express** como framework a utilizar, y **MongoDB** como base de datos, posee las siguientes ventajas/desventajas:

Ventajas :

- Se utiliza el mismo lenguaje de programación, tanto para frontend como para backend.
- Es un framework muy conocido y utilizado², por lo que cuenta con una amplia comunidad de desarrolladores.
- Es una solución con amplio soporte, por lo que puede ser desplegada en multitud de plataformas y servicios de cloud computing.

Ventajas :

- Simplicidad en la estructura, los documentos se almacenan en formato JSON, lo que permite una fácil integración con el frontend.
- Al ser una base de datos No-SQL, posee mayor flexibilidad a la hora de definir la estructura de los documentos⁵.
- Es una base de datos muy conocida y utilizada, por lo que cuenta con una amplia documentación⁶.

Desventajas/preocupaciones :

- No existe un orden específico para la creación de la aplicación de express, es tarea del desarrollador mantener una estructura y orden determinado.
- No es un framework orientado a la seguridad, por lo que es necesario implementar soluciones de seguridad adicionales³.
- Para mantener una escalabilidad aceptable, es esencial mantener una estructura de código ordenada y definida, utilizando patrones de diseño y buenas prácticas⁴.

Desventajas :

- Al no ser de tipo relacional, no es la alternativa más eficiente para calculos matemáticos sobre los datos⁵.
- Al no poseer un esquema predefinido, su flexibilidad puede provocar que los datos no estén bien estructurados.
- Es necesario definir una estructura, y mantenerla a través de validaciones al insertar nuevos datos, para mantener escalabilidad.

¹<https://docs.aws.amazon.com/apigateway/latest/developerguide/http-api.html>

²Dato de descargas semanales - web NPM: <https://www.npmjs.com/package/express>

³<https://expressjs.com/es/advanced/best-practice-security.html>

⁴<https://www.geeksforgeeks.org/model-view-controllermvc-architecture-for-node-applications/>

⁵<https://www.ibm.com/cloud/blog/sql-vs-nosql>

⁶<https://www.mongodb.com/docs/>

🏠 Arquitectura web

La arquitectura de desarrollo del proyecto web es de 3 capas, las que se presentan a continuación:

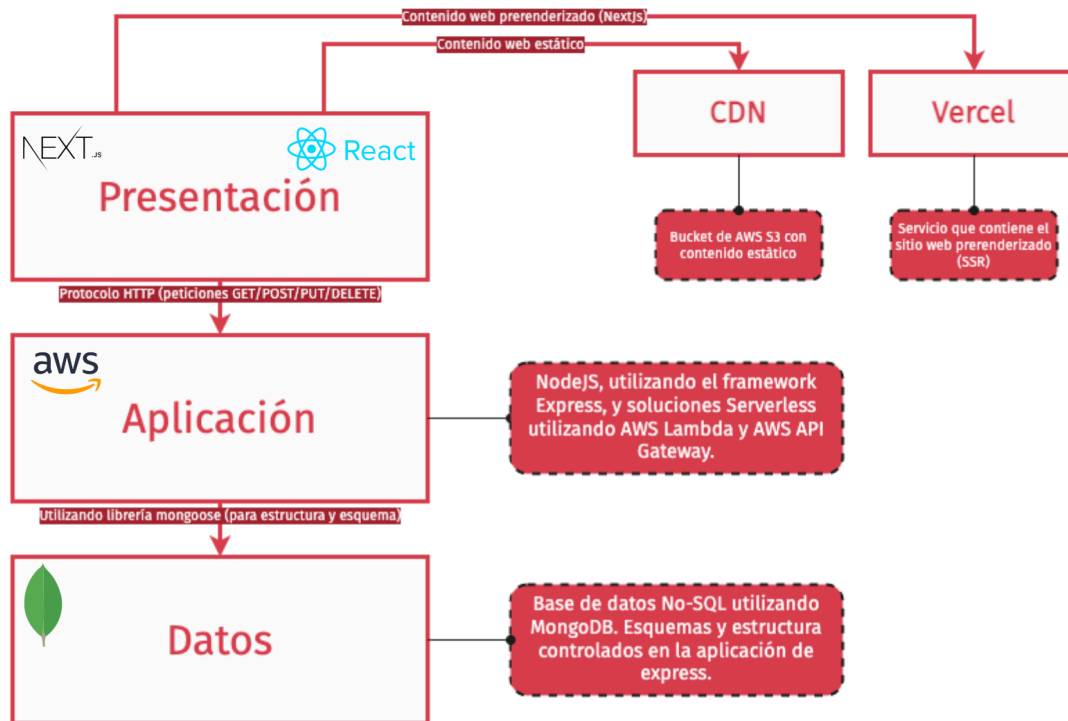


Figura 1: Arquitectura web de 3 capas - Proyecto Mapler

- | | | |
|---------------------|---|---|
| Presentación | : | Corresponde a la capa que se encarga de mostrar la aplicación al usuario. Al ser una aplicación web, podrá ser accedida a través de cualquier agente web (como por ejemplo, navegadores). El contenido del frontend será alojado tanto en una CDN (bucket de 📁 AWS S3), para almacenar datos estáticos como logotípos, imágenes, videos, y en el servicio ▲ Vercel, que provee almacenamiento estático del sitio web prerenderizado, para mejorar tiempos de carga y análisis de SEO ⁷ . |
| Aplicación | : | Capa donde se encuentra la aplicación de express, que se comunica con el frontend a través de servicios HTTP, y contendrá las rutas necesarias para el despliegue de la información requerida. La aplicación se encuentra en servicios <i>sin servidor</i> , alojados en usando 🌌 Lambda. |
| Datos | : | Capa donde se encuentran los datos, almacenados en una base de datos MongoDB alojada en 💎 MongoDB Atlas. |