TALLER HTTP - PARTE 1 - Desarrollo web

-Comparativo arquitecturas CLEAN, MVC, Tres Capas, Microservicios

|  |  |  |  |
| --- | --- | --- | --- |
| CLEAN | MVC | Tres Capas | Microservicios |
| Es la forma de ordenar y proyectar nuestro código hacia el futuro, para que al añadir nuevas características y resolver errores sea lo mas simple posible con la menor cantidad de desarrolladores.  Clean Architecture, termino dado por Rober C. Martin o mejor conocido como tío Bob. EL recopilo los modelos de capas mas utilizados en una versión mejorada llamándola Clean Architecture.  tío Bob recomienda estructurar u ordenar nuestro código en 3 capas principales:   1. Presentation: Interface de Usuario UI. 2. Domain: Todas las reglas de negocio, también se llama business logic en inglés. 3. Data: Todo lo que tenga que ver con la persistencia de datos.   Princicpios de Clean Architecture.   1. Es independiente de cualquier framework 2. Testeable 3. Independiente de la interfaz de usuario (UI) 4. Independiente de la base de datos 5. Independiente de cualquier elemento externo   https://platzi.com/clases/1619-arquitectura-android/21324-que-es-clean-architecture/ | MVC significa modelo (model) vista (view) controlador (controller). Esto es lo que significan cada uno de esos componentes.   * Modelo: El backend que contiene toda la lógica de datos * Vista: El frontend o interfaz gráfica de usuario (GUI) * Controlador: El cerebro de la aplicación que controla como se muestran los datos.   El concepto de MVC fue introducido por primera vez por Trygve Reenskaug, quien lo propuso como una forma de desarrollar el GUI de aplicaciones de escritorio.  Hoy en día, el patrón MVC se utiliza para aplicaciones web modernas porque permite que la aplicación sea escalable, mantenible y fácil de expandir.  El patrón MVC te ayuda a dividir el código frontend y backend en componentes separados. De esta manera, es mucho más fácil administrar y hacer cambios a cualquiera de los lados sin que interfieran entre sí.  https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/ | Se define a la arquitectura de tres capas como organizar el modelo en capas, lo que quiere decir que los componentes de una capa solo pueden hacer referencia a componentes en capas inferiores, esto simplifica la comprension y la organizacion del desarrollo de sistemas complejos, reduciendo las dependencias de capas mas bajas.  También ayuda a identificar que puede reutilizarse lo que proporciona una estructura que nos ayuda a tomar decisiones sobre que partes comprar y que partes construir.  Lo que al final la comunidad de software creo la noción de una arquitectura de tres niveles.  Lo que hace que la aplicación se divida en tres capas lógicas distintas, cada una estará con un grupo de interfaces permanentemente definido.  La primera capa se denomina capa de presentación y normalmente consiste en una interfaz gráfica de usuario de algún tipo.  La capa intermedia, o capa de empresa, consiste en la aplicación o lógica de empresa, y la tercera capa, la capa de datos, contiene los datos necesarios para la aplicación.  Esta separación entre la lógica de aplicación de la interfaz de usuario añade una enorme flexibilidad al diseño de la aplicación.  https://www.ecured.cu/Arquitectura\_de\_tres\_niveles#Arquitectura\_de\_tres\_niveles | Los microservicios son pequeños e independientes, y están acoplados de forma imprecisa. Cada servicio es un código base independiente por lo que solo puede administrarse por un equipo pequeño de desarrollo. Los servicios son los responsables de conservar sus propios datos o estado externo. Esto difiere del modelo tradicional, donde una capa de datos independiente controla la persistencia de los datos, estos servicios se comunican entre si a través de una API bien definidas. Los detalles de la implementación interna de cada servicio se ocultan frente a otros servicios y también admite la programación políglota. Por ejemplo, no es necesario que los servicios compartan la misma pila de tecnología, las bibliotecas o los marcos.  Además de los propios servicios, hay otros componentes que aparecen en una arquitectura típica de microservicios:  **Administración e implementación.**  **Puerta de enlace de API**  Entre las ventajas de usar una puerta de enlace de API se encuentran las siguientes:  Desacoplan los clientes de los servicios. Los servicios pueden cambiar de versión o refactorizarse sin necesidad de actualizar todos los clientes.   * Los servicios pueden utilizar los protocolos de mensajería que no son fáciles de usar para un servicio web, como AMQP. * La puerta de enlace de API puede realizar otras funciones transversales como la autenticación, el registro, la terminación SSL y el equilibrio de carga. * Directivas estándar, como, por ejemplo, para la limitación, el almacenamiento en caché, la transformación o la validación.   Tiene ventajas como:  **Agilidad**  **Equipos pequeños y centrados**  **Base de código pequeña**  **Mezcla de tecnologías**  **Aislamiento de errores**  **Escalabilidad**  **Aislamiento de los datos**  https://docs.microsoft.com/es-es/azure/architecture/guide/architecture-styles/microservices |

- Explicación de los protocolos http y https

HTTP: es un protocolo de comunicación que define la manera en que se comunican un dispositivo cliente y un servidor conectados a través de la web, HTTP es el idioma en que hablan los dispositivos informáticos para entenderse con las paginas web. Los dispositivos utilizan el protocolo HTTP cuando se conectan a cualquier web, lo que permite que el navegador pueda acceder a una pagina web cuando se ingresa a la dirección en la barra de búsqueda. También es el protocolo que permite descifrar la información de la web que se visita y mostrarla en pantalla. Sin embargo, no se encuentra protegido de forma segura, así que las webs que utilizan el protocolo HTTP no cuentan con un certificado SSL.

HTTP significa Hyper Text Transfer Protocol (Protocolo de Transferencia de Hipertexto). Es el protocolo utilizado por tus dispositivos para comunicarse con las páginas web, pero no se encuentra protegido de forma segura.

HTTPS: Es la evolución del protocolo HTTP diseñada para garantizar la seguridad de las comunicaciones en HTTP con el protocolo SSL/TLS (Secure Socket Layer / Transport Layer Security) para otorgarle al protocolo HTTP una capa de protección que no tiene en su versión básica.

HTTPS surgió en 1995, cuando internet comenzó a transformarse en un espacio más complejo donde los usuarios podían intercambiar información sensible que era necesario proteger, como sus datos bancarios o sus claves de acceso personales a diferentes plataformas. Gracias a esta evolución pudieron surgir servicios como la banca online o el e-commerce. Las webs que utilizan HTTPS pueden contar con un certificado SSL.

HTTPS significa Hyper Text Transfer Protocol Secure (Protocolo Seguro de Transferencia de Hipertexto). En esencia, este protocolo es una versión mejorada y segura del protocolo HTTP.

HTTP vs HTTPS

El navegador alternara entre ambos protocolos al navegar por internet, ya que son utilizados por un gran nummero de paginas web que se visitan a diario, y, en materia de transferencia de datos, no presentan apenas diferencias más allá de su seguridad.

Pero cada vez es de mayor importancia la seguridad en internet lo que ha hecho en los últimos años que el protocolo HTTPS sea casi esencial para la navegación, ya que el protocolo HTTP deja expuesta los datos privados. Lo que a día de hoy el certificado SSL es importante a la hora de valorar la seguridad de una web siendo así que la mayoría de los navegadores negara el acceso a paginas web que no cuenten con dicha certificación.

https://nordvpn.com/es/blog/diferencia-entre-http-y-https/

- Cuales son los puertos de http y https. Explique que son puertos

un puerto es el destino de la comunicación. Los puertos están designados por números, por debajo de 1024 cada puerto está asociado por defecto con un protocolo específico.

El puerto por defecto para el protocolo HTTP es 80 y el puerto por defecto para el protocolo HTTPS es 443, por lo tanto un servidor HTTP espera peticiones en esos puertos. Cada protocolo de internet está asociado con un puerto por defecto: SMTP (25), POP3 (110), IMAP (en-US) (143), IRC (194), y así sucesivamente.

Puerto 80: Este puerto es el que se usa para la navegación web de forma no segura HTTP

Puerto 443: Este puerto es también para la navegación web, pero en este caso usa el protocolo HTTPS que es seguro y utiliza el protocolo TLS por debajo.

- Qué son los headers http

Las cabeceras HTTP son la parte central de esas solicitudes y respuestas HTTP, y transportan información sobre el navegador cliente, la página solicitada, el servidor y más.

Las Cabeceras pueden ser agrupadas de acuerdo a sus contextos:

* Cabecera general: Cabeceras que se aplican tanto a las peticiones como a las respuestas, pero sin relación con los datos que finalmente se transmiten en el cuerpo.
* Cabecera de consulta: Cabeceras que contienen más información sobre el contenido que va a obtenerse o sobre el cliente.
* Cabecera de respuesta: Cabeceras que contienen más información sobre el contenido, como su origen o el servidor (nombre, versión, etc.).
* Cabecera de entidad: Cabeceras que contienen más información sobre el cuerpo de la entidad, como el tamaño del contenido o su tipo MIME.

- Qué son los métodos o verbos HTTP

HTTP define un conjunto de métodos de petición para indicar la acción que se desea realizar para un recurso determinado. Aunque estos también pueden ser sustantivos, estos métodos de solicitud a veces son llamados HTTP verbs. Cada uno de ellos implementan una semántica diferente, pero algunas características similares son compartidas por un grupo de ellos

Entre ellos los que métodos están son:

* GET
* HEAD
* POST
* PUT
* DELETE
* CONNECT
* OPTIONS
* TRACE
* PATCH

https://developer.mozilla.org/es/docs/Web/HTTP/Methods

- Qué son los HTTP status code

Los códigos de estado de respuesta HTTP indican si se ha completado satisfactoriamente una solicitud HTTP específica. Las respuestas se agrupan en cinco clases:

1. Respuestas informativas (100–199),
2. Respuestas satisfactorias (200–299),
3. Redirecciones (300–399),
4. Errores de los clientes (400–499),
5. y errores de los servidores (500–599).

- Comparativo servidores web Apache vs Nginx

|  |  |
| --- | --- |
| Apache | Nginx |
| Apache HTTP Server es un software de servidor web gratuito y de código abierto para plataformas Unix con el cual se ejecutan el 46% de los sitios web de todo el mundo. Es mantenido y desarrollado por la Apache Software Foundation  Apache es uno de los servidores web más populares que te permite ejecutar un sitio web seguro sin demasiados dolores de cabeza. Es la opción más común de los emprendedores en solitario y pequeñas empresas que desean tener presencia en la web.  **Pros:**   1. De código abierto y gratuito, incluso para uso comercial. 2. Software confiable y estable. 3. Parches de seguridad regulares y actualizados con frecuencia. 4. Flexible debido a su estructura basada en módulos. 5. Fácil de configurar para principiantes. 6. Multiplataforma (funciona tanto en servidores Unix como en Windows). 7. Viene listo para trabajar con sitios de WordPress. 8. Enorme comunidad y soporte fácilmente disponible en caso de cualquier problema.   **Contras:**   1. Problemas de rendimiento en sitios web con demasiado tráfico. 2. Demasiadas opciones de configuración pueden generar vulnerabilidades de seguridad. | NGINX, pronunciado en inglés como «engine-ex», es un famoso software de servidor web de código abierto. En su versión inicial, funcionaba en servidores web HTTP. Sin embargo, hoy en día también sirve como proxy inverso, balanceador de carga HTTP y proxy de correo electrónico para IMAP, POP3 y SMTP. NGINX ofrece una arquitectura asíncrona y controlada por eventos, característica que hace de NGINX uno de los servidores más confiables para la velocidad y la escalabilidad.  **Ventajas de usar Nginx:**   * Se trata de un **software multiplataforma,** por lo tanto, podremos instalarlo en la mayoría de nuestros servidores. * Consume **menos recursos** que la mayoría de los servicios que hacen su misma función. * Nos proporciona un **alto rendimiento** soportando **mayor carga** y respondiendo mejor que sus competidores. * Puede ser usado como **proxy inverso** cacheando el contenido de nuestros sitios web. * Podemos integrarlo junto con **Apache,**de forma que Nginx procese contenido estático y Apache contenido dinámico. * Puede usarse como balanceador de carga entre varios servidores, permitiéndonos así una mayor facilidad a la hora de escalar nuestros servidores. * Es compatible con una gran variedad de CMS y aplicaciones actuales como pueden ser: Wordpress, Drupal, Prestashop, y muchas más. * El proyecto Nginx tiene detrás a la empresa Nginx Inc. y también cuenta con el apoyo de una gran comunidad contribuyendo a mejorar y resolver dudas. También podemos recurrir a soporte profesional.   **Inconvenientes de usar Nginx:**   * No se integra con PHP de forma nativa. Es necesario usar FastCGI para poder integrarlo. |

- V8 engine relación con los navegadores

V8 es un motor de código abierto para JavaScript creado por Google. Cada navegador tiene su propio motor, pero con la evolución rápida de V8, otros navegadores han optado por integrarlo. Por este motivo, Microsoft Edge u Opera han decidido migrar a V8 para mejorar el rendimiento de su software.

¿Cuál es el objetivo de V8?

Este fue creado por la necesidad de Google de desarrollar su aplicación Google Maps de la manera más óptima posible, ya que los motores antiguos no permitían ejecutar aplicaciones con alta demanda computacional. Node.js utiliza el motor V8 para trabajar con JavaScript del lado del servidor de manera estable.

- Averigue los comandos PING,WHOIS, NSLOOKUP,TELNET, NETSTAT

PING: Ping es una herramienta de diagnóstico para **el símbolo del sistema deWindows**. El programa de líneas de comandos sirve, en el marco del diagnóstico de red, para establecer si se puede acceder a otro ordenador conectado en una red local o red pública.

WHOIS: es un directorio público mediante el cual puede saber «quién es» («who is» en inglés) el propietario de un dominio o dirección IP. Antes podía utilizar WHOIS para conocer algunos datos del propietario, como su nombre, dirección, número de teléfono y dirección de correo electrónico.

NSLOOKUP: nslookup es una herramienta de línea de comandos muy práctica y fácil de usar, cuya función básica es encontrar la dirección IP de un equipo determinado o realizar una búsqueda DNS inversa (es decir, encontrar el nombre de dominio de una determinada dirección IP). nslookup se ejecuta en la interfaz de líneas de comando del sistema operativo correspondiente: mientras que los usuarios de Windows iniciarán el servicio a través del símbolo del sistema (CMD), los de Unix lo hacen a través del terminal. Además, en la actualidad existen varios servicios web que permiten usar nslookup también online.

TELNET: permite el control remoto de los ordenadores por medio de entradas y salidas basadas en texto. Con este objetivo, se crea una conexión cliente-servidor a través del protocolo TCP y del puerto TCP 23, donde el dispositivo controlado ejerce de servidor y espera a los comandos pertinentes.

NETSTAT: término derivado de “network” (red) y “statistics” (estadísticas), es un programa dirigido con órdenes ejecutadas en la línea de comandos que entrega estadísticas básicas sobre la totalidad de las actividades de red. También puede entregar información acerca de los puertos y direcciones a través de los cuales se ejecutan las conexiones TCP y UDP, al igual que los puertos abiertos para solicitudes.

<https://www.ionos.es/digitalguide/servidores/herramientas/comando-ping/>

https://help.one.com/hc/es/articles/115005595885--Qu%C3%A9-es-WHOIS-

<https://www.ionos.es/digitalguide/servidores/herramientas/nslookup/>

<https://www.ionos.es/digitalguide/servidores/herramientas/telnet-el-protocolo-para-cualquier-plataforma/>

https://www.ionos.es/digitalguide/servidores/herramientas/una-introduccion-a-netstat/

- Construya una prćtica donde se pueda aplicar los conceptos estudiados.

PARTE 2 NODE-JS

* 1. Qué son módulos en Javascript

A partir de ECMAScript (ES2015) se introduce una característica nativa denominada Módulos ES (ESM), que permite la importación y exportación de fragmentos de datos entre diferentes ficheros Javascript, eliminando las desventajas que teníamos hasta ahora y permitiendo trabajar de forma más flexible en nuestro código Javascript.

https://lenguajejs.com/javascript/modulos/que-es-esm/

* 1. Cómo se utiliza el módulo fs para leer de un archivo

En NodeJS todas las operaciones de acceso al sistema de archivos están englobadas dentro del módulo "fs" (File System). Si queremos leer un archivo de texto que tenemos en local simplemte usaremos ese módulo para extraer el contenido del fichero, indicando su ruta y otra serie de parámetros

https://desarrolloweb.com/articulos/lectura-archivos-nodejs.html

* 1. Qué es JSON y cómo se construye información en formato JSON.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

* Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
* Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o sequencias.

https://www.json.org/json-es.html

4. Realice un programa para leer datos de un archivo con información en JSON y recorrerlo con un ciclo for (Use el archivo adjunto example.json)

5. Que es npm y cómo se usa

npm es el Node Package Manager que viene incluido y ayuda a cada desarrollo asociado a Node.

Esta herramienta funciona de dos formas:

* Como un repositorio ampliamente utilizado para la publicación de proyectos Node.js de código abierto. Lo que significa que es una plataforma en línea donde cualquiera puede publicar y compartir herramientas escritas en JavaScript.
* Como una herramienta de línea de comandos que ayuda a interactuar con plataformas en línea, como navegadores y servidores. Esto ayuda a instalar y desinstalar paquetes, gestión de versiones y gestión de dependencias necesarias para ejecutar un proyecto.

https://www.hostinger.co/tutoriales/que-es-npm

1. Averigüe que es nodemon y como se pueden ejecutar aplicaciones Javascript usandolo

**Nodemon** es una utilidad que monitorea los cambios en el código fuente que se esta desarrollando y automáticamente reinicia el servidor. Es una herramienta muy útil para desarrollo de aplicaciones en nodejs.

1. Cómo es la estructura de un proyecto node.js. Cómo se debe configurar si lo quiero ejecutar con nodemon.

https://medium.com/williambastidasblog/estructura-de-una-api-rest-con-nodejs-express-y-mongodb-cdd97637b18b

1. Instalar MongoDB.
2. Crear una base de datos con MongoDB.
3. Instalar las dependencias (Fichero de configuración principal package.json).
4. Hacer la conexión a MongoDB desde NodeJS con Mongoose. (en el index.js).
5. Crear el servidor web con NodeJS. API RESTful en Node.js (en el app.js).
6. Cargar el fichero app.js con la configuración de Express (en el index.js).
7. Crear los esquemas y modelos.
8. Crear un controlador.
9. Crear la ruta para el controlador.
10. Importar y cargar las rutas en el servidor (app.js).
11. Testear mi api con Postman.

Configuración: https://medium.com/javascript-espa%C3%B1ol/configuraci%C3%B3n-de-nodemon-en-un-servidor-node-js-6fde6275b50f