UNIVERSIDAD TÉCNICA NACIONAL

SEDE DEL PACÍFICO

INGENIERIA EN TECNOLOGÍAS DE LA INFORMACIÓN

TECNOLOGÍAS Y SISTEMAS WEB ll

SEBASTIÁN ALBERTO MATA ORTEGA

WALTER ALONSO ALPÍZAR ARGÜELLO

ll CUATRIMESTRE

2023



**INTRODUCCIÓN**

Dentro de este documento encontraremos información relevante acerca del curso de Tecnologías y Sistemas Web ll, donde nos enfocaremos en la parte del back-end empleando diferentes tecnologías que nos facilitarán desarrollar aplicativos webs, velar por la seguridad e integridad del usuario que utiliza el aplicativo, los diferentes formatos de intercambio de información, la manera en la que se envían, procesan y se acceden los datos. De esta manera los usuarios que necesiten de un servicio puedan tener acceso a la información del servicio que brinda el aplicativo web.

**Semana 1: Tecnologías para creación de API’s**

**Introducción a Frameworks y otras tecnologías**

**JavaScript & Frameworks**

JavaScript es un lenguaje de programación tipado de manera dinámica, el cual se ejecuta en el lado del cliente y permite interactuar con los elementos de la página web.

**Características:**

-**Interactividad en tiempo real**: JavaScript permite que las páginas web respondan a las acciones del usuario en tiempo real.

-**Versatilidad**: : Javascript es un lenguaje muy versátil que se puede utilizar tanto en el lado del cliente como en el lado del servidor.

-**Facilidad de aprendizaje**: Javascript es relativamente fácil de aprender para aquellos que ya tienen conocimientos de programación.

-**Amplia comunidad y Recursos**: Javascript cuenta con una comunidad activa de desarrolladores en todo el mundo.

**TypeScript**

Este es un superconjunto de javascript el cual es tipado de manera estática, en donde las variables que se establezcan van a trabajar como el tipo de variable que se le indique; INT, FLOAT, STRING, etc.

**Herramientas y Frameworks populares**

Hoy en día existen distintos tipos de tecnologías las cuales nos permiten juntar los diferentes algoritmos y servicios que elaboremos, específicamente dentro del ecosistema de **JavaScript** encontramos muchos **frameworks**, los cuales nos permiten realizar aplicaciones web más complejas e intuitivas.

Algunos de los más utilizados son:

* React: Biblioteca especializada en construir interfaces de usuario dentro del lenguaje de JS.
* Angular: Enfocado en la creación de aplicativos webs en JS.
* Vue.js: Framework progresivo para construir interfaces de usuario.
* Node.js: Entorno de ejecución de JavaScript de parte del servidor.
* Express: Tiene como objetivo crear aplicaciones web y APIs en Node.js.

**Ventajas de TypeScript sobre JavaScript**

* Detección de errores temprana: El tipado estático de TypeScript permite detectar errores de tipo durante la etapa de desarrollo, lo que ayuda a evitar errores comunes antes de ejecutar el código.
* Mantenibilidad y escalabilidad: TypeScript facilita el mantenimiento y la evolución de proyectos a largo plazo, especialmente en aplicaciones complejas, gracias a su soporte para clases, interfaces y otras características de programación orientada a objetos.
* Mejor herramienta de desarrollo: TypeScript proporciona una experiencia de desarrollo más rica, con autocompletado, sugerencias de errores y refactorización de código más avanzada en comparación con JavaScript puro.
* Compatibilidad con JavaScript: Dado que TypeScript es un superconjunto de JavaScript, se puede utilizar en proyectos existentes, ya que todo el código JavaScript válido también es válido en TypeScript.

**Node.js**

Este es un entorno de tiempo de ejecución (runtime) de JS basado en el motor V8 de Chrome. A diferencia de JavaScript, el cual se ejecuta sobre el navegador, node.js es ejecutado sobre la parte del servidor, ya que Node.js permite que las funciones de JS puedan interactuar con el servidor para realizar todo tipo de tareas. De igual manera, este posee un modelo de programación basado en eventos y en I/O sin bloqueo, permitiéndole manejar diversas solicitudes de manera simultáneamente sin bloquear el hilo principal de ejecución. Otras características de Node.js son las siguientes:

1. **Amplio ecosistema de módulos y paquetes:** Se cuenta con un gestor de paquetes llamado **npm (Node Package Manager)**, el cual permite gestionar una amplia variedad de módulos.
2. **Desarrollo rápido y productivo:** Permite un desarrollo efectivo y rápido debido a la ligeres por naturaleza y a su capacidad para ejecutar varias tareas en paralelo.
3. **Escalabilidad y rendimiento:** El hecho que permite desarrollar diferentes tareas en paralelo y al modelo sin bloqueo influye en el hecho que le permite a las empresas instalarla dentro de sistemas muy concurrentes por una gran cantidad de usuarios.

**Angular**

Este es un Framework de desarrollo de aplicaciones web, basado en TypeScript, creado por Google. Proporciona un conjunto de herramientas que nos permiten crear aplicativos webs de mayor magnitud, escalables y de alto rendimiento.

**Características Principales.**

Angular posee **Componentes y Directivas** el cual se ayuda para poder crear la arquitectura basada en componentes, en donde cada **componente** es un bloque de construcción importante para la aplicación. Cada componente tiene su propio código. De igual manera Angular ofrece un poderoso sistema de **Binding** que permite mantener sincronización entre los datos y la interfaz de usuario. Así mismo, Angular cada que ves que se necesita crear una nueva instancia este posee **Inyección de dependencias** con el objetivo de reutilizar código, evitando el agotamiento en memoria, y finalmente tenemos que angular como tal posee una función de **Enrutamiento**, el cual nos permite establecer una navegación entre los diferentes componentes y vistas del trabajo al definir rutas y asociar componentes a estas rutas.

**TypeOrm**

TypeOrm es un ORM que puede correr sobre Node.js, el navegador, Ionic, entre otros. Tiene como objetivo brindar soporte las características más recientes de JS, a su ves que provee nuevas características que nos ayudarán en el desarrollo de las aplicaciones.

**Características**

TypeOrm posee un Soporte multiplataforma: TypeORM es compatible con una variedad de bases de datos relacionales, como **MySQL, PostgreSQL, SQLite, Microsoft SQL Server y Oracle**. Esto permite a los desarrolladores elegir la base de datos que mejor se adapte a sus necesidades. TypeORM facilita la comunicación entre la base de datos y los objetos de la aplicación mediante el **mapeo de tablas y columnas** de la base de datos a clases y propiedades de TypeScript, TypeORM proporciona herramientas para crear y **modificar esquemas de base de datos** de forma programática. También admite migraciones de esquemas, lo que facilita la actualización y mantenimiento de la estructura de la base de datos a medida que la aplicación evoluciona.

Algunas otras Características:

* **Consultas y Relaciones**
* **Soporte para transacciones y caching**
* **Integración con Express y otros framworks**

**MySQL**

MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) muy popular y ampliamente utilizado. Es de código abierto y está basado en el modelo de datos relacional, lo que significa que almacena datos en tablas estructuradas con filas y columnas.

**Características**

1. **Escalabilidad y Rendimiento:** MySQL está diseñado para manejar grandes volúmenes de datos y es altamente escalable. Puede manejar múltiples conexiones y consultas concurrentes.
2. **Lenguaje SQL:** : MySQL utiliza el lenguaje SQL (Structured Query Language) como su lenguaje de consulta principal. SQL proporciona una sintaxis estándar para interactuar con la base de datos
3. **Soporte Multiplataforma:** : MySQL es compatible con una amplia gama de plataformas, incluyendo Windows, macOS y varias distribuciones de Linux
4. **Alta disponibilidad y replicación :** MySQL ofrece opciones para alta disponibilidad y replicación de datos. Mediante la configuración de réplicas, puedes crear copias de tus datos y distribuir la carga entre varios servidores.

**Semana 2: Proyecto de Typeorm**

Durante esta clase se aprendió acerca de como podemos modificar nuestro proyecto hecho en typescript para poder elaborar un aplicativo web de manera que atienda las necesidades del servicio que estemos brindando. Para realizar la creación de nuestro api, vamos a utilizar diferentes tecnologías, como las son:

* **TypeOrm:** Es una herramienta la cual nos permite realizar conexiones entre las tecnologías que vayamos a emplear, en este caso Node.js, TypeScript, express, entre otras.
* **Express:** Framework que posee conexiones con Node.js para realizar las funciones respectivas a nuestro aplicativo web.
* **TypeScript:** Lenguaje de tipo tipado estático, que nos servirá para poder realizar las diferentes consultas y funciones.
* **Node.JS:** Trabaja como backend y proporciona todas las funciones que van detrás del servidor.

Los pasos para crear nuestro proyecto son:

1. Instalar Node.js en nuestro dispositivo
2. Abrir ventana de cmd para instalar los paquetes correspondientes.
3. “CMD”: npm install typeorm –save
4. “CMD”: npm install reflect-metada –save
5. “CMD”: npm install @types/node --save-dev
6. “CMD”: npm install mysql –save
7. “CMD”: npx typeorm init --name MyProject --database postgres
8. “CMD”: cd MyProject
9. “CMD”: npm install
10. “CMD”: npm install express –save
11. “CMD”: npm start
12. “CMD”: code . (Para ejecutar nuestro proyecto dentro de visual Studio

Ahora que tenemos nuestro proyecto montado deberíamos de ver algo así:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteA partir de aquí cuando tenemos nuestro proyecto de prueba podemos establecer nuestra propia conexión hacia nuestra base de datos preferida dentro de nuestro paquete de data.source:

Podemos modificar los parámetros de nuestra base de datos correspondiente. Para que nuestro proyecto cumpla nuestras necesidades, debemos tener en cuenta que nuestro api se maneja mediante rutas, las cuales nos permitirán extraer la información en base a las rutas o caminos donde estarán los métodos que harán los debidos request (get, post, put or delete). Donde dichos request o funciones van a estar preparados dentro de un archivo controlador que nos brindará la información con base a la ruta que se requiere.

**Semana 3: CRUD Methods**

Durante esta semana se realizaron los debidos métodos mediante los cuales vamos a realizar diferentes acciones, ya sea: mostrar la información general, mediante un filtro de ID, Insertar información, actualizar la información y borrarla. Debemos de tener en cuenta las necesidades de nuestro servicio. En el proyecto pasado creamos las rutas mediante las cuales vamos a acceder a las diferentes acciones:

Texto

Descripción generada automáticamente

Cada acción de http lleva consigo una ruta especifica y en algunos casos, trae consigo un identificador de búsqueda único para realizar acciones específicas. Cabe Recalcar que estos métodos apuntan hacía un controlador del cliente, en donde se encuentran todas las acciones.

Cada uno de estos hace una conexión hacia la base de datos creando una constante la cual va a obtener la información de la base de datos que estemos empleando, para después buscar la información dentro de ella y almacenarla dentro de otra constante.



Texto

Descripción generada automáticamenteDespués de buscar la información requerida podemos realizar diferentes validaciones con respecto a la base de datos, y a la información que se está intentando mandar o validar los datos que se brindan para poder realizar una búsqueda filtrada.

Cabe recalcar que estas validaciones traen consigo ciertos mensajes ya sea de representar errores o cuando se ejecutan las acciones como deberían de ser. Ya que se obtiene la información mediante un método get, cuando se requiere insertar nueva información se deben hacer muchas más validaciones, las cuales llamamos: **“Reglas de entrada”**, después de esto se validan lo que viene siendo las **reglas de negocio**, las cuales son propias de cada empresa con la que se vaya a trabajar. Para poder insertar dicha información debemos de mandar los detalles de lo que necesitamos en una manera concreta, ya que esto se trabaja mediante archivos de **JSON**, y dentro del código descomponer el objeto de JSON en pequeños pedazos para poder realizar las necesidades; ya sea: realizar las validaciones individuales o la inserción como tal de dicha información.