

Desafío - Citas médicas

- Para realizar este desafío debes haber estudiado previamente todo el material disponibilizado correspondiente a la unidad.
- Una vez terminado el desafío, comprime la carpeta que contiene el desarrollo de los requerimientos solicitados y sube el `.zip` en el LMS.
- Puntaje total: 10 puntos
- Desarrollo desafío:
 - El desafío se debe desarrollar de manera Grupal

Capítulos

El desafío está basado en los siguientes capítulos de la lectura:

- Conociendo NPM
- Manejo de dependencias
- Conociendo paquetes en Node
- Paquetes NPM que no procesan datos
- Paquetes NPM para el procesamiento de datos

Descripción

La clínica DENDE Spa ha tenido un problema con su software de gestión de citas y necesita urgentemente una aplicación capaz de registrar a los nuevos usuarios almacenando su nombre, apellido, hora en la que fue registrado e indispensablemente un código identificador.

Deberás crear un servidor que disponibilice una ruta para la consulta de todos los usuarios registrados.

1. Nombre: Aaron - Apellido: Martin - ID: 5373a7 - Timestamp: September 29th 2020, 8:52:54 pm
2. Nombre: Isabella - Apellido: Rasmussen - ID: 0c6059 - Timestamp: September 29th 2020, 8:53:24 pm
3. Nombre: Laica - Apellido: Hubert - ID: c51d45 - Timestamp: September 29th 2020, 8:53:26 pm
4. Nombre: Alvaro - Apellido: Pereira - ID: cf459d - Timestamp: September 29th 2020, 8:53:29 pm
5. Nombre: Batur - Apellido: TuǺǻlu - ID: e076f8 - Timestamp: September 29th 2020, 8:53:32 pm
6. Nombre: Aapo - Apellido: Harju - ID: db0295 - Timestamp: September 29th 2020, 8:53:34 pm
7. Nombre: Kiara - Apellido: Vincent - ID: ca07c3 - Timestamp: September 29th 2020, 8:53:37 pm

Requerimientos

1. El registro de los usuarios debe hacerse con la [API Random User](#) usando axios para consultar la data. **(2 Puntos)**
2. Cada usuario registrado debe tener un campo id único generado por el paquete UUID. **(2 Puntos)**
3. Cada usuario debe tener un campo timestamp almacenando la fecha de registro obtenida por medio del paquete Moment. **(2 Puntos)**
4. Por cada consulta realizada al servidor, se debe devolver al cliente una lista con los datos de todos los usuarios registrados usando Lodash para recorrer el arreglo de usuarios. **(2 Puntos)**
5. En cada consulta también se debe imprimir por la consola del servidor la misma lista de usuarios pero con fondo blanco y color de texto azul usando el paquete Chalk. **(1 Punto)**
6. El servidor debe ser levantado con el comando Nodemon. **(1 Punto)**