| **Laboratorio 6** |
| --- |
| **Lanzamiento de la Actividad**   * **Abra** el proyecto SD-12-1: AssessmentServer-Individual en VS code. * **Examine** el fragmento de código de ejemplo proporcionado. |
| **Fragmento de código de ejemplo**   | fetch("https://example.com/api", {  method: "POST",  body: JSON.stringify({  id: 6,  first\_name: "Alicia",  last\_name: "Miranda"  }),  headers: {  "Content-Type": "application/json; charset=UTF-8"  }  }) | | --- |   La API Fetch es una de las formas en que podemos realizar solicitudes web en JavaScript.   * Podemos especificar un método HTTP, como "POST". * Podemos usar JSON.stringify() para enviar datos JSON en un formato que el servidor entenderá. * Podemos incluir un encabezado "Content-Type" para que el servidor de la API sepa cómo interpretar los datos como JSON. |
| **Tareas**  Una API web permite que nuestras aplicaciones se comuniquen, soliciten **recursos** y reciban **respuestas** a través de Internet.  **Para su evaluación final de módulo, escribirá sus propios métodos para interactuar con una API web.**  Esta API web servirá una base de datos JSON que contiene una lista de usuarios. Cada usuario tendrá un número de identificación (**id**), un **Nombre**, un **Apellido** y una dirección de **correo electrónico** asociados. **Aprenderá por si mismo** cómo interactuar con un módulo **npm** llamado json-server.  **Tarea 1**: Cree y exporte un método para **devolver** la URL del servidor desde el proyecto SD-12-1: AssessmentServer-Individual   * El método getServerURL() debería **devolver** la URL de su servidor JSON en **funcionamiento**.   **Tarea 2**: Cree y exporte un método para imprimir una **lista** de usuarios desde el servidor JSON.   * El método listUsers() debería simplemente **imprimir** la respuesta JSON completa, que contiene la **lista** de usuarios, en la consola.   **Tarea 3**: Cree y exporte un método para **agregar** un nuevo usuario al servidor JSON.   * El método addUser() debe:   + Tomar **tres** entradas de la siguiente manera: addUser(<primer\_nombre>, <apellido>, <correo\_electrónico>)   + Resultar en la adición de un nuevo usuario completo al servidor JSON **con un nuevo número de identificación único, secuencial**.     - Por ejemplo, si el número de identificación más alto en la lista existente es 4, entonces este nuevo número de identificación debería ser 5.   **Tarea 4**: Cree y exporte un método para **eliminar** a un usuario del servidor JSON.   * El método delUser() debería:   + Tomar un número de identificación (**id**) como entrada de la siguiente manera: delUser(<id>)   + Resultar en que el usuario coincida con ese número de identificación (**id**) siendo eliminado del servidor JSON. |
| **Puntuación**  El instructor puntuará su evaluación al final de la sesión. |