IMPLEMENTACIÓN DE API BACKEND NODE EXPRESS

INSTRUCCIONES

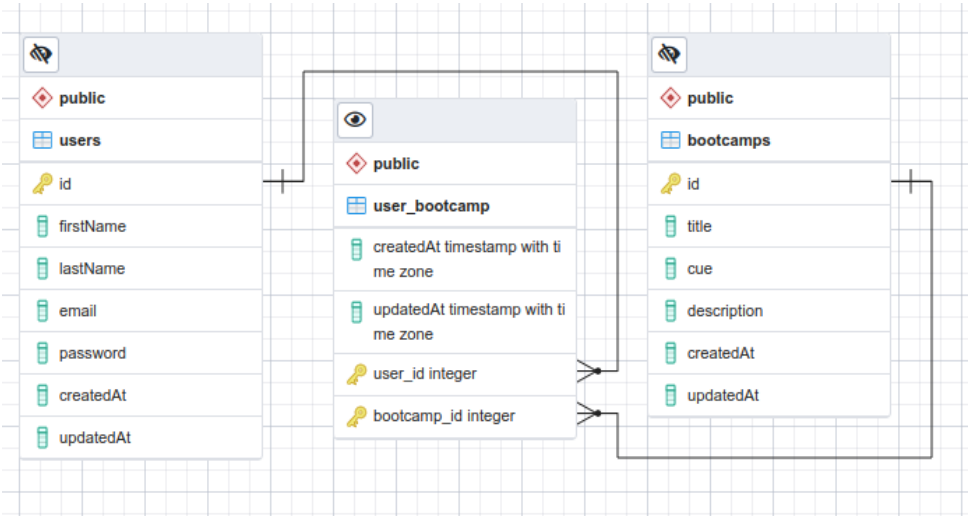
Lee cuidadosamente cada uno de los requerimientos que se te presentan a continuación y desarrolla la prueba de acuerdo con lo solicitado.

DESCRIPCIÓN:

El equipo de desarrollo de software ha comenzado un desarrollo para el acceso a datos por medio de una aplicación realizada por medio de Node.js.

Dicho desarrollo se basó en el diseño de acceso a datos por medio de sequelize y las relaciones respectivas para la gestión de cursos Bootcamp de una determinada empresa de adiestramiento. El equipo aplica la metodología scrum y ya realizó el primer Sprint del proyecto que se trató en el diseño e implementación por medio de Node.js para el registro de cursos Bootcamp y de usuarios de los mismos.

Ahora bien para éste segundo Sprint, se desea adecuar dicho diseño con la finalidad que éste disponible a través de una API RESTful, para éste nuevo sprint, se agrega a la tabla users el nuevo campo password, con un mínimo de 8 caracteres de requerimiento, con la finalidad de poder autenticar al usuario, el modelo de entidad relación de la base de datos es el que muestra a continuación:



El requerimiento emitido por la empresa de adiestramiento parte del principio de que los usuarios pueden participar en distintos bootcamp, y a su vez distintos bootcamp poseen distintos usuarios como se realizó en el primer sprint .

Para el segundo Sprint se desea la construcción de la API RESTful con Express del bootcamp, soportará el token basado en la autenticación con JWT(JSONWebToken) y PostgreSQL.

Para la construcción de la API debe contener los siguientes funcionalidades:

- Un usuario de puede registrar en la API
- Un usuario inicia sesión con el email y el password
- Los registros se guardarán en la base de datos PostgreSQL
- Una vez registrado el usuario usuario puede agregar bootcamp
- Puede asignar usuarios a los bootcamp
- La consulta de los bootcamp es pública

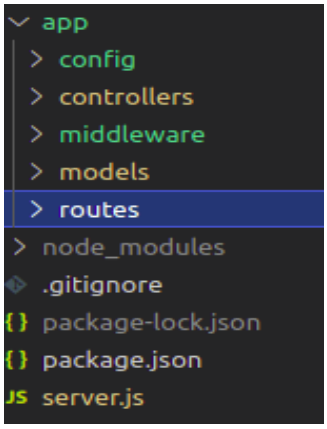
La APIs debe proveer las siguientes endpoint:

Métodos	URL	Acción
POST	/api/signup	Registro de una nuevo usuario, acceso público
POST	/api/signin	Inicio de sesión en la API, acceso público
GET	/api/user/:id	Lista información del usuario según id, acceso por medio de token, previamente iniciado sesión
GET	/api/user/	Lista información de todos los usuarios y los Bootcamp registrados, acceso por medio de token, previamente iniciado sesión

PUT	/api/user/:id	Actualiza los campos de firstName y lastName de un usuario según su id, acceso por medio de token, previamente iniciado sesión
DELETE	/api/user/:id	Elimina el usuario según id, acceso por medio de token, previamente iniciado sesión
POST	/api/bootcamp	Crea un bootcamp, acceso por medio de token, previamente iniciado sesión
POST	/api/bootcamp/adduser	Agrega usuarios previamente registrados al bootcamp, acceso por medio de token, previamente iniciado sesión
GET	/api/bootcamp/:id	Obtiene información de un bootcamp según id, y muestra los usuarios registrados en el bootcamp. Acceso por medio de token, previamente iniciado sesión
GET	/api/bootcamp	Lista todos los bootcamp, acceso público

En este segundo sprint, el equipo de desarrollo conjuntamente con el ScrumMaster acordaron lo siguientes pasos para el desarrollo del mismo:

1. Adecuar el proyecto y el código que se realizó en el primer sprint, el cual se adjunta el archivo llamado: `sprint_01_bootcamp.zip`, con la finalidad de que se adecue, mejoré y construya una API RESTful para el bootcamp según las rutas antes mencionadas.
2. Se implementa haciendo uso de las siguientes dependencias: express, sequelize, pg, pg-hstore, body-parser, cors, JWT(jsonwebtoken) y bcryptjs
3. Partir de la siguiente estructura para el desarrollo



4. Crear dentro de la carpeta `config`, el archivo `db.config.js`, que posee la configuración a la base de datos, el nombre de la base de datos es: `db_jwtbootcamp` y `auth.config.js` que contendrá la frase secreta para la creación del token respectivamente

5. Dentro de la carpeta `models`, se encuentran los modelos tanto para el usuario (`user.model.js`) como para el bootcamp (`bootcamp.model.js`). El archivo `index.js` se define la conexión con sequelize a la base de datos y modelos

6. En la carpeta `controllers` posee los controladores tanto para el usuario (`user.controller.js`), como para el bootcamp (`bootcamp.controller.js`).

   Para el usuario se deben adecuar los siguientes controladores para la API:

   - Crear y guardar usuarios llamado `createUser`
   - Obtener los bootcamp de un usuario llamado `findUserById`
   - Obtener todos los Usuarios incluyendo los bootcamp llamado `findAll`
   - Actualizar usuario por Id llamado `updateUserById`
   - Eliminar un usuario por Id llamado `deleteUserById`

   Para el bootcamp se debe adecuar los siguientes controladores para la API:

   - Crear y guardar un nuevo bootcamp llamado `createBootcamp`

- Agregar un Usuario al Bootcamp llamado `addUser`
- Obtener los bootcamp por id llamado `findById`
- Obtener todos los Usuarios incluyendo los Bootcamp llamado `findAll`

7. En la carpeta middleware contiene los siguientes archivos:

   `auth.js:` que contiene una función de verificar token llamada `verifyToken`

   `verifySingUp.js:` que verifica si el correo ya se encuentra ingresado al momento de registrarse un nuevo usuario

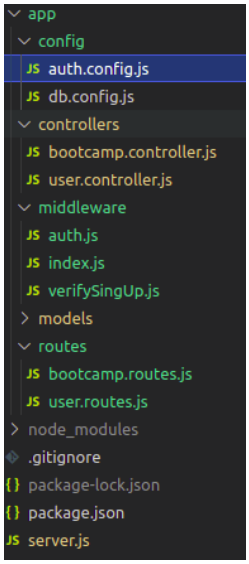
   `index.js:` módulo que exporta los middleware

8. En la carpeta routes contiene la definición de las rutas en los siguientes archivos:

   `user.routes.js:` definen las rutas de los usuarios

   `bootcamp.user.js:` definen las rutas para los bootcamp

   La estructura final es la siguiente:

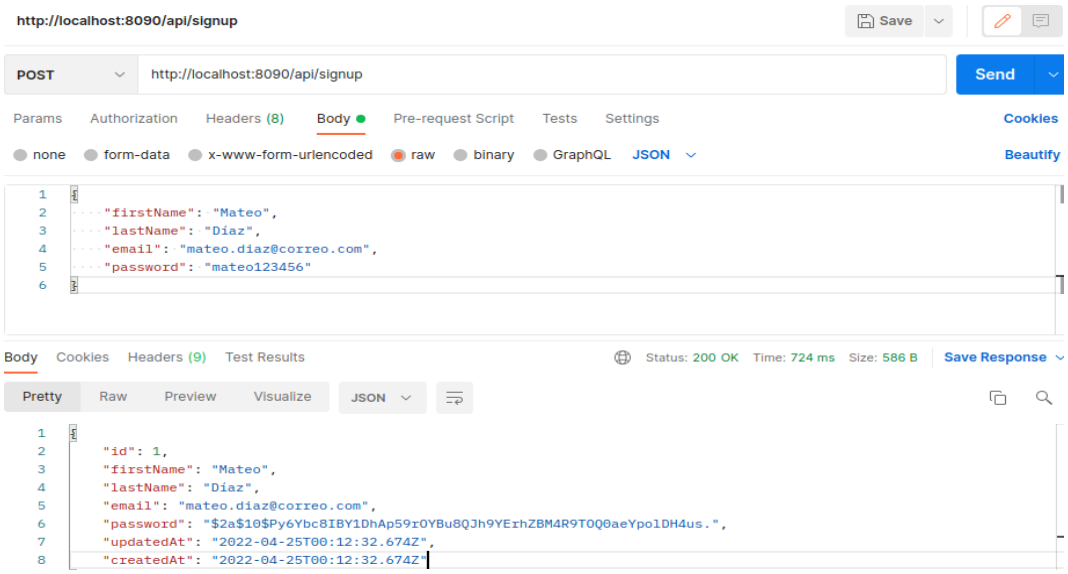


9. Finalmente para verificar los modelos y las relaciones con sus métodos y los endpoint se hace uso de la herramienta Postman .

   Crear los siguientes usuarios:

firstName	**lastName**	**email**	**password**
Mateo	Díaz	mateo.diaz@correo.com	mateo123456
Santiago	Mejías	santiago.mejias@correo.com	santiago123456
Lucas	Rojas	lucas.rojas@correo.com	lucas123456
Facundo	Fernández	facundo.fernandez@correo.com	facundo123456

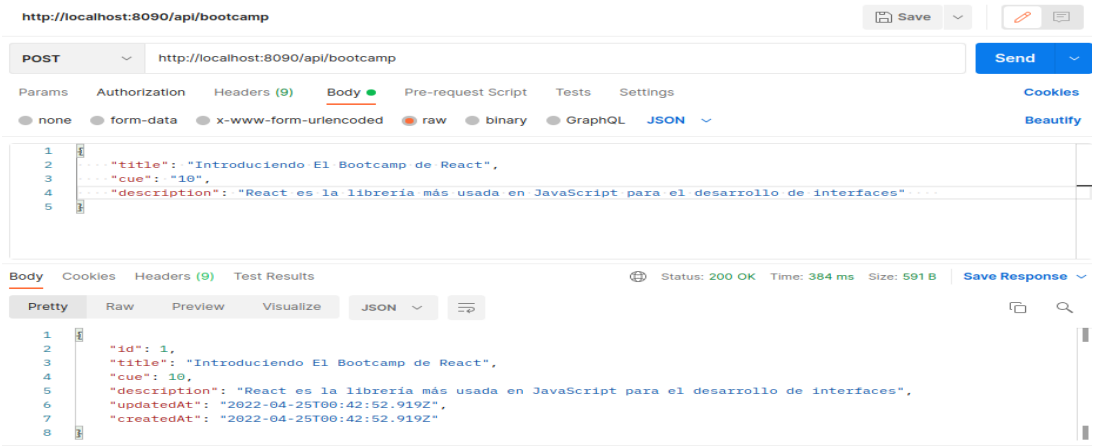
   Al ejecutar la creación del primer usuario, con postman obtenemos:



Crear los siguientes bootcamp:

title	**cue**	**description**
Introduciendo El Bootcamp de React	10	React es la librería más usada en JavaScript para el desarrollo de interfaces
Bootcamp Desarrollo Web Full Stack	12	Crearás aplicaciones web utilizando las tecnologías y lenguajes más actuales y populares como JavaScript, nodeJS, Angular, MongoDB, ExpressJS
Bootcamp Big Data, Inteligencia Artificial & Machine Learning	18	Domina Data Science todo el ecosistema de lenguajes y herramientas de Big Data e integrarlos con modelos avanzados de Artificial Intelligence y Machine Learning

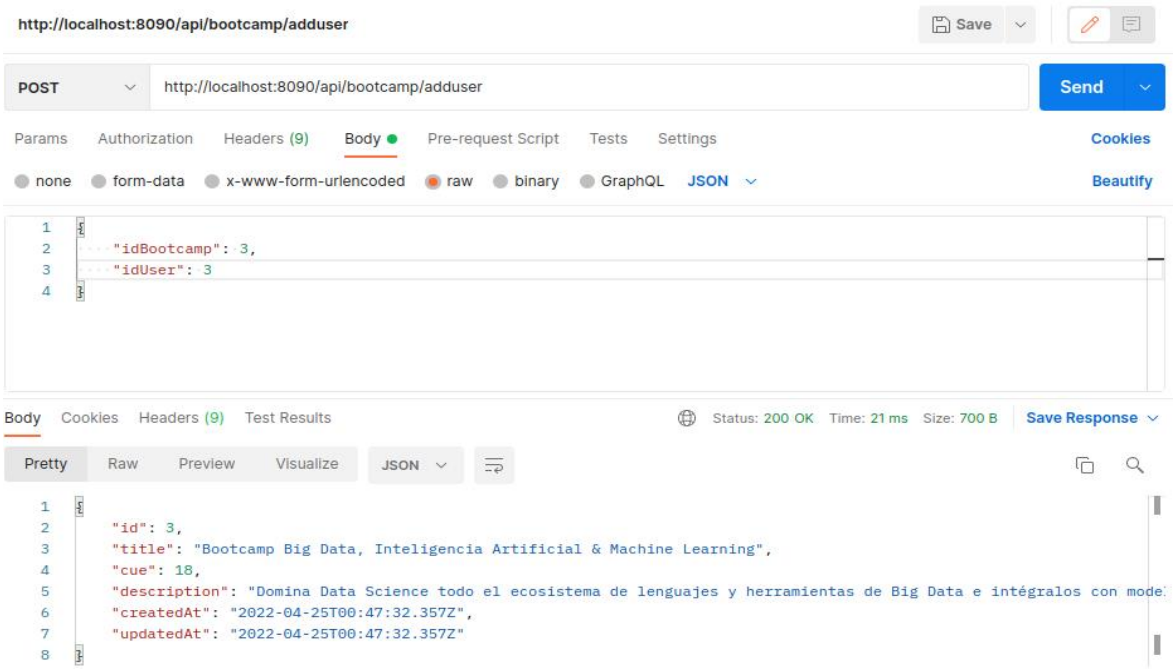
Al ejecutar la creación del primer bootcamp, observamos con postman lo siguiente:



Agregar los siguientes usuarios al bootcamp:

title	**usuarios**
Introduciendo El Bootcamp De React	Mateo Díaz Santiago Mejias
Bootcamp Desarrollo Web Full Stack	Mateo Díaz
Bootcamp Big Data, Inteligencia Artificial & Machine Learning	Mateo Díaz Santiago Mejias Lucas Rojas

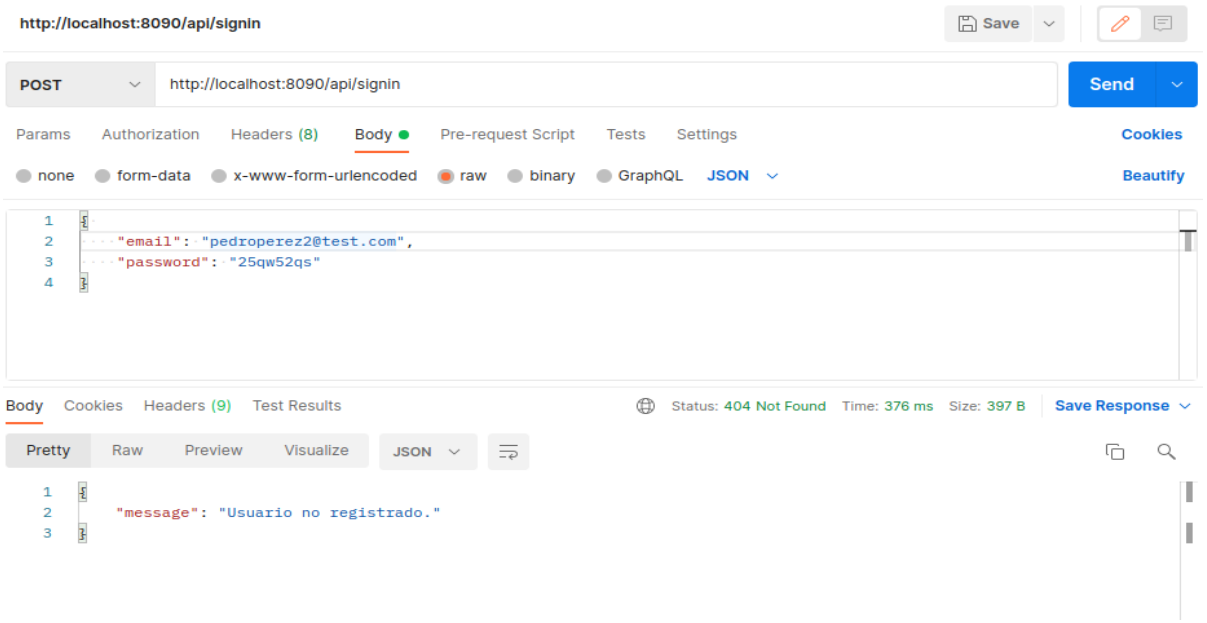
Al ejecutar la asignación de usuarios a los bootcamp, observamos con Postman lo siguiente:



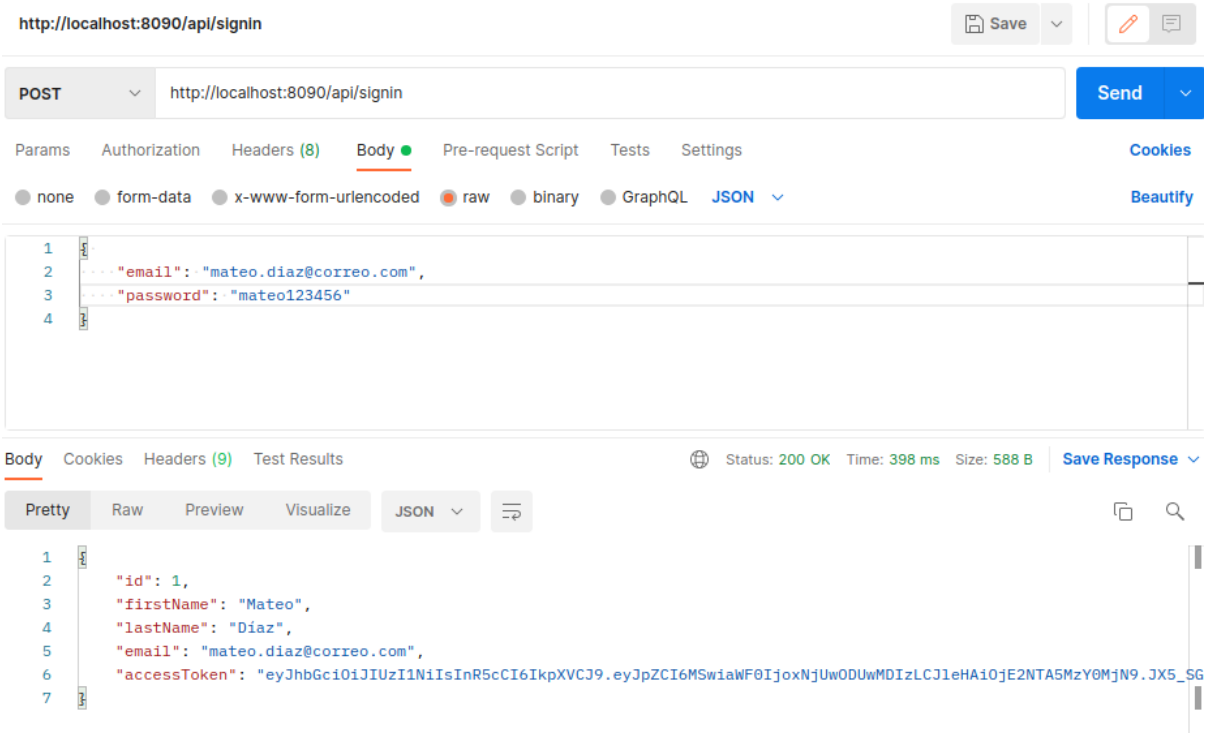
Realizar las siguientes consultas con Postman:

- Iniciar sesión de un usuario con email y contraseña, por ejemplo:

```
{
    "email": "pedroperez2@test.com",
    "password": "25qw52qs"
}
```



- Iniciar sesión de un usuario con email y contraseña, con usuario registrado



- Listar todos los usuarios con sus bootcamp

```
[{
        "id": 1,
        "firstName": "Mateo",
        "lastName": "Díaz",
        "email": "mateo.diaz@correo.com",
        "password":
"$2a$10$Py6Ybc8IBY1DhAp59rOYBu8QJh9YErhZBM4R9TOQ0aeYpolDH4us.",
        "createdAt": "2022-04-25T00:12:32.674Z",
        "updatedAt": "2022-04-25T00:12:32.674Z",
        "bootcamps": [{
                "id": 1,
                "title": "Introduciendo El Bootcamp de React"
            },
            {
                "id": 2,
                "title": "Bootcamp Desarrollo Web Full Stack"
            },
            {
                "id": 3,
                "title": "Bootcamp Big Data, Inteligencia Artificial
& Machine Learning"
            }
        ]
    },
    {
        "id": 2,
        "firstName": "Santiago",
        "lastName": "Mejías",
        "email": "santiago.mejias@correo.com",
        "password":
"$2a$10$GGuFx6AS4GrzKjCSJSbfs.XZIaQhXuDp5kaf.OqrYRKsFIXewdCpW",
        "createdAt": "2022-04-25T00:15:06.706Z",
        "updatedAt": "2022-04-25T00:15:06.706Z",
        "bootcamps": [{
                "id": 1,
                "title": "Introduciendo El Bootcamp de React"
            },
            {
                "id": 3,
                "title": "Bootcamp Big Data, Inteligencia Artificial
& Machine Learning"
            }
        ]
    },
    {
        "id": 3,
        "firstName": "Lucas",
```

```
        "lastName": "Rojas",
        "email": "lucas.rojas@correo.com",
        "password":
"$2a$10$aeeYHDZTsCs.9iagUgkQF.NlZGZvd4O7YBpC2apPn8vjHi7DpfxI2",
        "createdAt": "2022-04-25T00:16:11.347Z",
        "updatedAt": "2022-04-25T00:16:11.347Z",
        "bootcamps": [{
            "id": 3,
            "title": "Bootcamp Big Data, Inteligencia Artificial &
Machine Learning"
        }]
    }
]
```


- Listar el usuario con el id 3
- Actualizar el usuario según su id, por ejemplo actualizar el usuario con id=1 por Pedro Sánchez
- Eliminar un usuario por id, por ejemplo el usuario con id=1
- Consultando el bootcamp por id, incluyendo los usuarios registrados
- Listar todos los bootcamp con sus usuarios
- Consultar un usuario por id incluyendo los bootcamp
- Gestione adecuadamente el manejo de errores

Como referencia para la adecuación del controlador del usuario para el registro de un nuevo usuarios por medio de la ruta `/api/signup`, se puede tener presente lo siguiente:

```
exports.createUser = (req, res) => {
  // Validar el request
  if (!req.body.firstName || !req.body.lastName || !req.body.email || !req.body.password) {
    res.status(400).send({
      message: "Los campos son requeridos y no vacios!"
    });
    return;
  }

  // Crear un usuario
  const user = {
    firstName: req.body.firstName,
    lastName: req.body.lastName,
    email: req.body.email,
    password: bcrypt.hashSync(req.body.password)
  };

  // Guardando un usuario en la base de datos
  User.create(user)
    .then(data => {
      res.send(data);
      //res.send("usuario Registrado satisfactoriamente")
    })
    .catch(err => {
      res.status(500).send({
        message: err.message || "Algun error ha ocurrido en la creación del usuario."
      });
    });
};
```