

Desarrollo web *Backend*

Programación y administración de redes - Práctica 1
Grado en Ingeniería Informática
Departamento de Informática. Universidad de Jaén

Nombres: _____

21/2/2022

Instrucciones

El objetivo es mantener una base de datos de los *tickets* de solicitud recibidos en una empresa de servicio técnico. La base de datos se almacenará en un archivo JSON con la siguiente estructura:

```
{
  "tickets": [
    {
      "id": "c621231b-b173-4d8d-a5c8-1f70d3f0c825",
      "cliente": "Zapatería Shoes",
      "solicitud": "Fallo al agregar productos al catálogo",
      "fecha": "2021-10-31",
      "completada": true
    },
    ...
  ]
}
```

El identificador se generará de forma automática y permitirá referirse de forma única a cualquier ticket registrado en el sistema.

Sigue los pasos indicados a continuación para construir el *backend* de una aplicación cuyo *frontend* se diseñará en las dos prácticas posteriores a esta. Entrega a través de la plataforma el archivo JSON con la base de datos y el archivo .js con la aplicación.

1.- Crea un directorio donde almacenarás los archivos de la aplicación. Comienza por crear en él el archivo `db.json` con un par de tickets de solicitudes de servicio.

2.- Instala **Insomnia** en tu ordenador y define los siguientes *endpoints* para la aplicación:

- a) `GET /tickets` - Obtener la lista de los tickets existentes en la base de datos.
- b) `GET /ticket/:id` - Obtener los datos de un ticket concreto cuyo identificador se facilita como parámetro.
- c) `POST /nueva` - Agregar un nuevo ticket. El identificador y fecha los establecerá el servidor y el campo `completada` tomará el valor `false`. Se devolverá el *ticket* una vez creado.
- d) `GET /actualizaticket/:id/:completada` - Actualizar el campo `completada` de un ticket con el valor recibido como parámetro. No devolverá información alguna al *frontend*.
- e) `GET /borraticket/:id` - Eliminar de la base de datos el ticket indicado. Devolverá la base de datos tras el borrado.
- f) `POST /editar` - Modificar los datos de un ticket. El identificador servirá para localizar el ticket cuyos campos se actualizan con los recibidos en esta petición.

3.- Crea en la carpeta del paso 1 el archivo `index.js`:

- a) Introduce en él las sentencias apropiadas para cargar los paquetes `express`, `uuid` y `fs`.
- b) Carga en una variable el contenido del archivo `db.json`.
- c) Configura `express` para tomar datos de formularios y quedar a la escucha.
- d) Guarda el archivo y ejecuta en la consola el comando `node index.js`. El servidor, aunque todavía no responde a peticiones, debería quedar a la escucha sin emitir error alguno.
- e) Pulsa **^C** para interrumpirlo y volver a la línea de comandos.

4.- Editar el archivo `index.js` para implementar la petición HTTP del paso 2.a:

- a) Responde a la petición `GET /tickets` devolviendo el contenido de la variable que cargaste en el paso 3b.
- b) Guarda los cambios y lanza el servidor según se indica en el paso 3d.
- c) Usa **Insomnia** para ejecutar la petición definida en el paso 2.a. Verifica que la respuesta obtenida es correcta.
- d) Vuelve a la línea de comandos e interrumpe el servidor con **^C**.

5.- Siguiendo el mismo procedimiento descrito en el apartado 4, implementa en `index.js` las peticiones **2b** a **2f**. Cada vez que implementes una verifica su funcionamiento. Una vez hayas finalizado tendrás el *backend* preparado.