

ALGORITMO LevantarServicioAPI

```
// Acción:  
// Inicia la aplicación y configura las rutas y el servidor para escuchar en  
el puerto especificado  
// Métodos implementados: use, listen
```

```
Crear una instancia de la aplicación Express  
Utilizar middleware para parsear JSON  
LeerLibrosDesdeArchivo()  
Definir ruta GET "/bookRequest" que responde con ObtenerLibros()  
Definir ruta PUT "/bookRequest/:id" que responde con ActualizarLibro()  
Definir ruta POST "/bookRequest/:id/reserva" que responde con  
RealizarReserva()  
Definir ruta DELETE "/bookRequest/:id/reserva" que responde con  
EliminarReserva()  
Iniciar el servidor en el puerto 3001
```

FIN ALGORITMO

SubProceso LeerLibrosDesdeArchivo()

```
// Acción:  
// Lee los libros desde el archivo data.json y asigna a la variable libros  
// Método implementado: readFile, parse
```

```
Leer archivo data.json  
Si hay un error entonces  
    Escribir "Error al leer el archivo data.json"  
Sino  
    Parsear el contenido del archivo a JSON  
    Asignar los libros y reservas del JSON a las variables libros y reservas  
FinSi
```

FinSubProceso

SubProceso ObtenerLibros()

```
// Acción:  
// Devuelve la lista de libros al cliente  
// Método implementado: get
```

```
Responder con libros en formato JSON
```

FinSubProceso

SubProceso ActualizarLibro()

```
// Acción:  
// Actualiza un libro con la información proporcionada por el cliente  
// Método implementado: put
```

```
Leer parámetros id, título y autor  
Para cada libro en la lista de libros hacer  
    Si el id del libro es igual al id proporcionado entonces  
        Actualizar el título y autor del libro  
    FinSi
```

FinPara

```
Escribir los libros actualizados en el archivo data.json  
Responder con "Libro actualizado exitosamente"
```

FinSubProceso

SubProceso RealizarReserva()

```
// Acción:  
// Realiza una reserva para un libro específico  
// Método implementado: post
```

```
Leer parámetro id  
Buscar el libro con el id proporcionado  
Si el libro existe entonces  
    Crear una reserva con un ID único y asociarla al libro  
    Añadir la reserva a la lista de reservas
```

```
        Escribir los libros y reservas actualizados en el archivo data.json
        Responder con "Reserva realizada exitosamente"
    Sino
        Responder con "Libro no encontrado"
    FinSi
FinSubProceso

SubProceso EliminarReserva()
    // Acción:
    // Elimina una reserva específica
    // Método implementado: delete

    Leer parámetro id
    Buscar la reserva con el id proporcionado
    Si la reserva existe entonces
        Eliminar la reserva de la lista de reservas
        Escribir los libros y reservas actualizados en el archivo data.json
        Responder con "Reserva eliminada exitosamente"
    Sino
        Responder con "Reserva no encontrada"
    FinSi
FinSubProceso
```