

# LISTA

---

- ☐ Ticket 1
- ☐ Ticket 2
- ☐ Ticket 3
- ☐ Ticket 4
- ☐ Ticket 5
- ☐ Ticket 6
- ☐ Ticket 7
- ☐ Ticket 8
- ☐ Ticket 9
- ☐ Ticket 10
- ☐ Ticket 11
- ☐ Ticket 12
- ☐ Ticket 13
- ☐ Ticket 14
- ☐ Ticket 15

## Link

[https://github.com/SebastianAMo/BOOTCAMP\\_FOREST\\_TAREAS/blob/main/TAREA\\_10/TAREA\\_10.md](https://github.com/SebastianAMo/BOOTCAMP_FOREST_TAREAS/blob/main/TAREA_10/TAREA_10.md)

## TICKET 1: Configuración del Proyecto y Dependencias

Se debe configurar el proyecto desde cero utilizando **npm** y asegurarse de instalar las dependencias necesarias, tanto para el funcionamiento del servidor como para el desarrollo (entorno de desarrollo). Las dependencias principales son: Express, dotenv, cors, morgan y nodemon.

Característica	Datos
Tiempo estimado	15 minutos
Lenguaje	JavaScript
OS	Windows 10 x64
Tipo de tarea	Configuración

## TICKET 2: Configuración de Variables de Entorno

Es necesario crear un archivo **.env.example** con las variables de entorno requeridas: **PORT**, **DATA\_PATH** y **NAME\_FILE**. Este archivo debe contener valores de ejemplo y una descripción de cada variable y los valores aceptados.

Característica	Datos
Tiempo estimado	5 minutos
Lenguaje	N/A

Característica	Datos
OS	N/A
Tipo de tarea	Configuración

## TICKET 3: Configuración Automatizada

Se debe crear un archivo `config.sh` que automatice la creación de la carpeta `current_status`, el archivo `.env` y la copia del contenido de `.env.example` al `.env`. Además, se debe agregar la carpeta `current_status` al archivo `.gitignore` si aún no está presente.

Característica	Datos
Tiempo estimado	15 minutos
Lenguaje	Bash Script
OS	Windows 10 x64
Tipo de tarea	Automatización

## TICKET 4: Configuración del Módulo Config

Crear la carpeta `config` y dentro de ella el archivo `config.js`. Este archivo debe utilizar la biblioteca `dotenv` para extraer las variables de entorno del archivo `.env`. Se debe definir un objeto `config` con las configuraciones leídas del archivo `.env` y exportarlo para que pueda ser importado en otras partes del proyecto.

Característica	Datos
Tiempo estimado	20 minutos
Lenguaje	JavaScript
OS	Windows 10 x64
Tipo de tarea	Desarrollo

## TICKET 5: Creación del Servidor

En el archivo `index.js`, se deben importar las dependencias necesarias y el módulo de configuración (`config.js`). Se debe configurar el servidor Express con CORS habilitado y usando el puerto especificado en `config.js`. Además, se deben agregar las rutas y middlewares necesarios para el funcionamiento del servidor.

Característica	Datos
Tiempo estimado	30 minutos
Lenguaje	JavaScript
OS	Windows 10 x64

Característica	Datos
Tipo de tarea	Desarrollo

## TICKET 6: Creación de Routers

Crear la carpeta `router` y dentro de ella los archivos: `index.js`, `movement.router.js`, `jump.router.js`, `attack.router.js` y `idle.router.js`. En los endpoints omite la parte de `http://localhost:3000/api/v1` que se muestra en la documentación, solo se define con lo ultimo.

Característica	Datos
Tiempo estimado	5 minutos
Lenguaje	JavaScript
OS	Windows 10 x64
Tipo de tarea	Desarrollo

## TICKET 7: Rutas y Endpoints Iniciales

Completar los endpoints en `movement.router.js`, `jump.router.js`, `attack.router.js` y `idle.router.js` para que devuelvan un mensaje indicando el nombre del endpoint al ser accedidos.

Todos se debe conectar en el `index.js` en una funcion `loadEndpoints` y que agregue la version de la api `/api/v1`

Cargar la funcion `loadEndpoint` en el `index.js` principal para que sean funcionales

Característica	Datos
Tiempo estimado	30 minutos
Lenguaje	JavaScript
OS	Windows 10 x64
Tipo de tarea	Desarrollo

## TICKET 8: Pruebas con Postman

Crear todos los endpoints en Postman y verificar que respondan correctamente con los mensajes esperados.

Característica	Datos
Tiempo estimado	30 minutos
Lenguaje	N/A
OS	N/A
Tipo de tarea	Pruebas

## TICKET 9: Creación de la Carpeta de Servicios

Crear la carpeta `service` y dentro de ella los archivos: `game.service.js` y `file.service.js`.

Característica	Datos
Tiempo estimado	5 minutos
Lenguaje	JavaScript
OS	Windows 10 x64
Tipo de tarea	Desarrollo

## TICKET 10: Implementación del Servicio de Archivos

En `file.service.js`, implementar los métodos `createFile`, `readFile` y `deleteFile` para manejar la creación, lectura y eliminación de archivos de forma asíncrona. Todos los métodos deben recibir el nombre del archivo, y `createFile` debe enviar los datos que se guardarán en el archivo.

Característica	Datos
Tiempo estimado	30 minutos
Lenguaje	JavaScript
OS	Windows 10 x64
Tipo de tarea	Desarrollo

## TICKET 11: Conexión de Servicio de Archivos con Servicio de Juego

En `game.service.js`, implementar los métodos `startGame`, `getGameStatus` y `deleteGameStatus` para utilizar los métodos del servicio de archivos y manejar el estado del juego. En el método `startGame`, el dato que se envía es la estructura en la documentación.

Conectar los servicios hechos en los endpoints del archivo `idle.router.js`

Característica	Datos
Tiempo estimado	40 minutos
Lenguaje	JavaScript
OS	Windows 10 x64
Tipo de tarea	Desarrollo

## TICKET 12: Actualización de Archivo del Estado del Juego

Implementar el método `updateFile` en `file.service.js` para actualizar un archivo de forma asíncrona. Utilizar el método `updateFile` en `game.service.js` en el método `updateGameStatus`, también de forma asíncrona.

Característica	Datos
Tiempo estimado	30 minutos
Lenguaje	JavaScript
OS	Windows 10 x64
Tipo de tarea	Desarrollo

## TICKET 13: Completar los Endpoints de Movement

Completar los endpoints en `movement.router.js` utilizando el método `updateGameStatus` para procesar las solicitudes de movimiento. Por ejemplo, los movimientos "up" y "right" aumentan en 10 en sus ejes correspondientes, mientras que "down" y "left" disminuyen en 10. Se debe obtener el estado actual y actualizarlo a partir de ese.

Característica	Datos
Tiempo estimado	40 minutos
Lenguaje	JavaScript
OS	Windows 10 x64
Tipo de tarea	Desarrollo

## TICKET 14: Completar los Endpoints de Jump

Completar los endpoints en `jump.router.js` utilizando el método `updateGameStatus` para procesar las solicitudes de salto. Si se salta hacia la derecha, el valor de "x" y "y" debe ser 10; para saltar hacia la izquierda, el valor de "x" debe ser -10 y "y" 10. En ambos casos, el valor de "jumping" debe ser "true".

Característica	Datos
Tiempo estimado	30 minutos
Lenguaje	JavaScript
OS	Windows 10 x64
Tipo de tarea	Desarrollo

## TICKET 15: Completar los Endpoints de Attack

Completar los endpoints en `attack.router.js` utilizando el método `updateGameStatus` para procesar las solicitudes de ataque. Cuando se ataca hacia la derecha, el valor de "attacking" debe ser 10; para atacar hacia la izquierda, el valor debe ser -10.

Característica	Datos
Tiempo estimado	35 minutos
Lenguaje	JavaScript

Característica	Datos
OS	Windows 10 x64
Tipo de tarea	Desarrollo