

# API Rest

**Juan José Coronado, Jose Moran,  
Jorge Sobrino y Carlos Soler**

# Índice

- Descripción del proyecto
- Forma de entrega
- Documentación para el funcionamiento
- Entorno utilizado
- Cronología
- Funcionamiento
- Demostración del funcionamiento

# Descripción del proyecto

La idea principal del proyecto era crear un servicio web para la gestión de eventos de ocio sobre el que se permitiese realizar distintos métodos.

Los métodos soportados son POST, GET, DELETE y PUT.

El servicio web permite tanto crear, obtener, actualizar y borrar eventos, lugares y organizadores de manera que el usuario lo pueda realizar de manera sencilla a través de una interfaz web , tal y como se mostrará en la demostración. De manera adicional se hace uso de una API ya creada para obtener información acerca del tiempo.

# Forma de entrega

Todo el código desarrollado ha sido subido a un repositorio de GITHUB, compartido con el profesor, y que ha permitido el desarrollo colaborativo de todos los integrantes del grupo.

El repositorio contiene todos los archivos necesarios para el correcto funcionamiento del servicio además de un README con la documentación necesaria para hacer funcionar el proyecto.

# Documentación para funcionamiento

El repositorio contiene todos los archivos necesarios para el funcionamiento del servicio web. Tras clonar el repositorio será necesario iniciar una base de datos en MongoDB utilizando los parámetros que se presentan en el fichero “enviroment”, siguiendo el modelo de la base de datos

Posteriormente se deberán instalar las dependencias mediante el comando *npm install* tanto dentro del servicio web como en la carpeta app-web-eventos

Finalmente se ejecutará un *npm start* dentro del proyecto y dentro del directorio app-web-eventos para arrancar el servicio web y el cliente. El servicio web se arranca en la dirección localhost:3000 y el cliente se arranca en localhost:5000, desde donde se podrán realizar las peticiones

# Entorno utilizado (I)

Principalmente se ha utilizado WebStorm como entorno de desarrollo por todos los integrantes del grupo. Se ha hecho uso de Node.js, entorno en tiempo de ejecución y multiplataforma y se ha usado GitHub como sistema de control de versiones.

Para la base de datos se ha elegido una base de datos no relacional, en este caso MongoDB, con distintas colecciones siendo estas **eventos**, **lugares** y **organizadores**. La colección **eventos** contiene los campos de título, descripción, duración, aforo, lugar y organizador, **lugares** está formado por nombre, ubicación y ciudad y **organizadores** se compone de nombre de empresa, país de origen y dueño

## Entorno utilizado (II)

Se ha creado una API propia a través de OpenAPI donde se definen todos los endpoints que contendrá la aplicación además de los métodos soportados por cada ruta y los códigos de respuesta esperados por cada una de ellas.

# Cronología del proyecto

## TASKS





# Funcionamiento

Tras realizar todos los pasos descritos anteriormente, en el apartado de documentación para el funcionamiento, será necesario conectarse a localhost:5000

Al conectarnos nos encontraremos con la página index desde donde podemos elegir ir a una categoría u otra, pudiendo elegir entre eventos, organizadores, lugares y tiempo

Para las tres primeras categorías podremos realizar los métodos GET, PUT, POST y DELETE donde para su funcionamiento únicamente será necesario rellenar los campos en cuestión deseados

Para la categoría de tiempo será necesario escribir la ubicación de donde queremos consultar el tiempo en cuestión

A continuación se realizará una demostración de todo lo comentado en esta diapositiva y en toda la presentación

# Demo!



¡Gracias!