

DOCUMENTACIÓN

El objetivo de esta aplicación será almacenar animales en una base de datos además de poder hacer ciertos cálculos con los datos de dichos animales. Recordamos que estos datos los extraeremos a través de una serie de métodos.

En cuanto a las aplicaciones usadas podemos ver que he subido la aplicación a un repositorio de la página web Github. Para poder subir aplicaciones a esta página web debemos de tener en cuenta los siguientes comandos:

En primer lugar tenemos que crear nuestro repositorio para esto usaremos el comando `git init`. Tenemos que tener en cuenta de que vamos a usar alguna herramienta de escritura para definir nuestros proyectos pero sin abusar de los comentarios, para esto usaremos el fichero `README.md`.

SUBIR EL PROYECTO A GITHUB

Ahora bien cuando todo esta listo siempre tendremos que usar los mismos comandos para comenzar con nuestro repositorio:

`git init -->` con esto damos constancia de que vamos a usar comando `git`

`git commit -m "mensaje" -->` con esto podremos comentar lo que queramos como subtítulo en el fichero y se verá reflejado en la página de github

`git branch -M master -->` Es como dar permiso de administración ya que será maestro

`git remote add origin enlace -->` esto sirve para indicar donde vamos a almacenar el proyecto y por eso tendremos que usar nuestro código propio.

`git push -u origin master`

IMPORTANTE existe una forma de hacer que no se suban las carpetas y esto será gracias a la carpeta `.gitignore` ya que no se subirán todas aquellas carpetas cuyo nombre haya sido indicado.

RESUBIR INFORMACIÓN MODIFICADA A GITHUB

Recordar rápidamente que para volver a subir información con cambios a github solo tendremos que usar los siguientes comandos:

`git init`

`git add .`

`git commit -m "mensaje"`

`git push -u origin master`

También recordar que tendremos que modificar el fichero tsconfig.json

```
"target": "es6", línea 7  
"outDir": "./build", línea 17
```

Cada vez que queramos ir probando todas las modificaciones que vamos haciendo en nuestra aplicación vamos a dejar un terminal compilando, para esto escribimos: “npm ts” y dejaremos el terminal en el modo node compilando esperando cambios en el código para irlos modificando. Cuando queramos iniciar la aplicación utilizaremos otro comando diferente que será: “npm start” Vemos que podemos usar esos comandos específicos porque en el fichero package.json escribimos las siguientes líneas de código para que reconozca los comandos.

```
"scripts": {  
  
  "ts": "tsc -w",  
  
  "dev": "nodemon ./build/server.js",  
  
  "start": "node ./build/server.js"  
},
```

Información extra para crear una RestApi

npm init -y → Para crear el package.json

→ instalar typescript local si no se instaló global

npm i typescript --save-dev *Local

npm i typescript -g *Global

→ Iniciar proyecto typescript

<https://medium.com/@bhagyamangale/tsc-init-4665ec9d7b09>

tsc --init //Si global. Se crea el tsconfig.json

npx tsc --init //Si local. Se crea el tsconfig.json

git init // Para crear el repositorio local

→ Instalación de express, mongoose y morgan

<https://dev.to/mtee/getting-started-with-morgan-3d1m>

→ la petición que hace el usuario (request) hasta que la petición llega al servidor

npm i express mongoose morgan

→ Instalamos los tipos de datos y módulos de desarrollo

```
npm install @types/node @types/mongoose @types/express @types/morgan nodemon typescript -D
```

→ Crear una nueva rama

```
git branch rama002
```

→ Cambiar de rama

```
git checkout rama002
```

```
git add .
```

```
git commit -m "en nueva rama"
```

```
git push -u origin rama002
```