## Log de Errores en SLACK

**U-TAD** 

## Monitorización: enviar errores a Slack

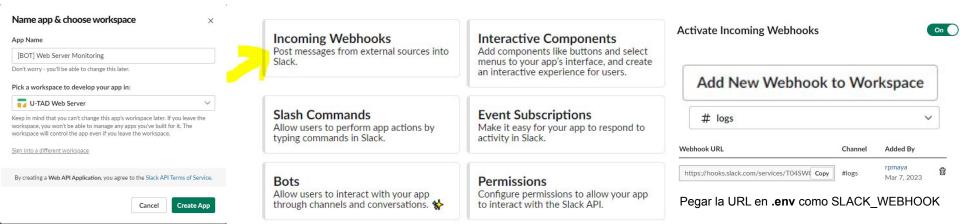
https://slack.com/ (abre una cuenta si no la tienes ya)

Crea un nuevo espacio de trabajo y dentro de Slack un Canal cerrado "logs"

Una vez creada la cuenta y el espacio de trabajo accedemos a:

https://api.slack.com/messaging/webhooks

En 1.- Pulsa el botón Create your Slack app (From Scratch)



## Monitorización: enviar errores a Slack

npm i morgan-body (https://www.npmjs.com/package/morgan-body)

})

npm i @slack/webhook (Sniffer de lo que pasa en nuestro Server y envío a Slack respectivamente). Modificamos app.js (después lo abstraeremos a distintos handlers): const morganBody = require("morgan-body") const {IncomingWebhook} = require("@slack/webhook") const webHook = new IncomingWebhook (process.env.SLACK WEBHOOK) const loggerStream = { write: message => { webHook.send({ text: message }) }, morganBody(app, { noColors: true, //limpiamos el String de datos lo máximo posible antes de mandarlo a Slack skip: function(req, res) { //Solo enviamos errores (4XX de cliente y 5XX de servidor) return res.statusCode < 400 }, stream: loggerStream

## Monitorización: enviar errores a Slack

Nos llevamos el objeto loggerStream desde app.js a utils/handleLogger.js (creadlo)

```
const { IncomingWebhook } = require("@slack/webhook")
const webHook = new IncomingWebhook(process.env.SLACK WEBHOOK)
const loggerStream = {
    write: message => {
       webHook.send({
         text: message
module.exports = loggerStream
Y en app.js:
const express = require("express")
const cors = require("cors")
const morganBody = require("morgan-body")
require("dotenv").config();
const loggerStream = require("./utils/handleLogger")
const dbConnect = require("./config/mongo")
morganBody(app, {
   noColors: true,
   skip: function(req, res) {
       return res.statusCode < 400
   stream: loggerStream
})
```