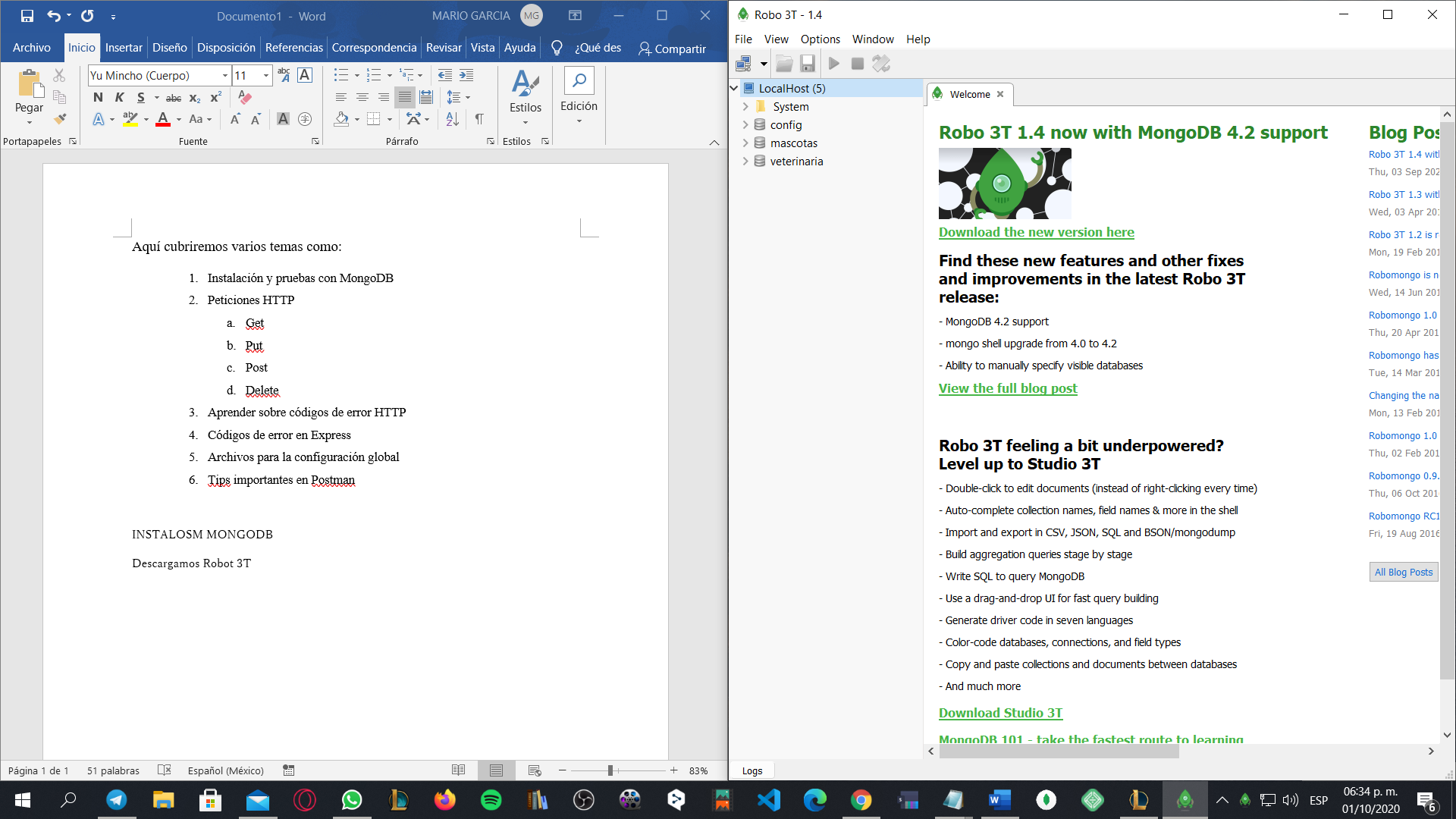
Aquí cubriremos varios temas como:

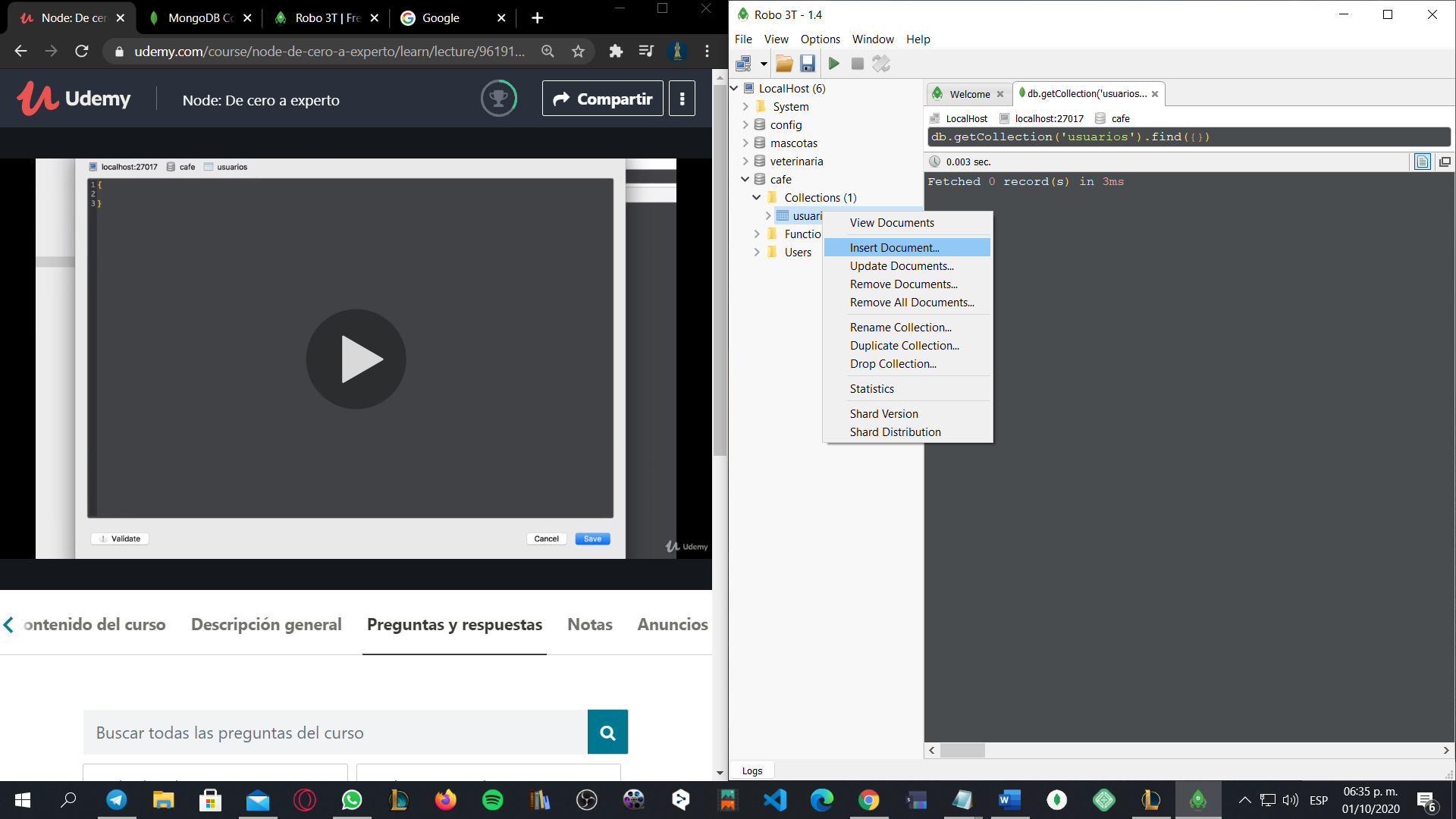
1. Instalación y pruebas con MongoDB
2. Peticiones HTTP
   1. Get
   2. Put
   3. Post
   4. Delete
3. Aprender sobre códigos de error HTTP
4. Códigos de error en Express
5. Archivos para la configuración global
6. Tips importantes en Postman

INSTALOSM MONGODB

Descargamos Robot 3T, creamos una nueva conexión en local y nos aparecen nuestra Bases de datos

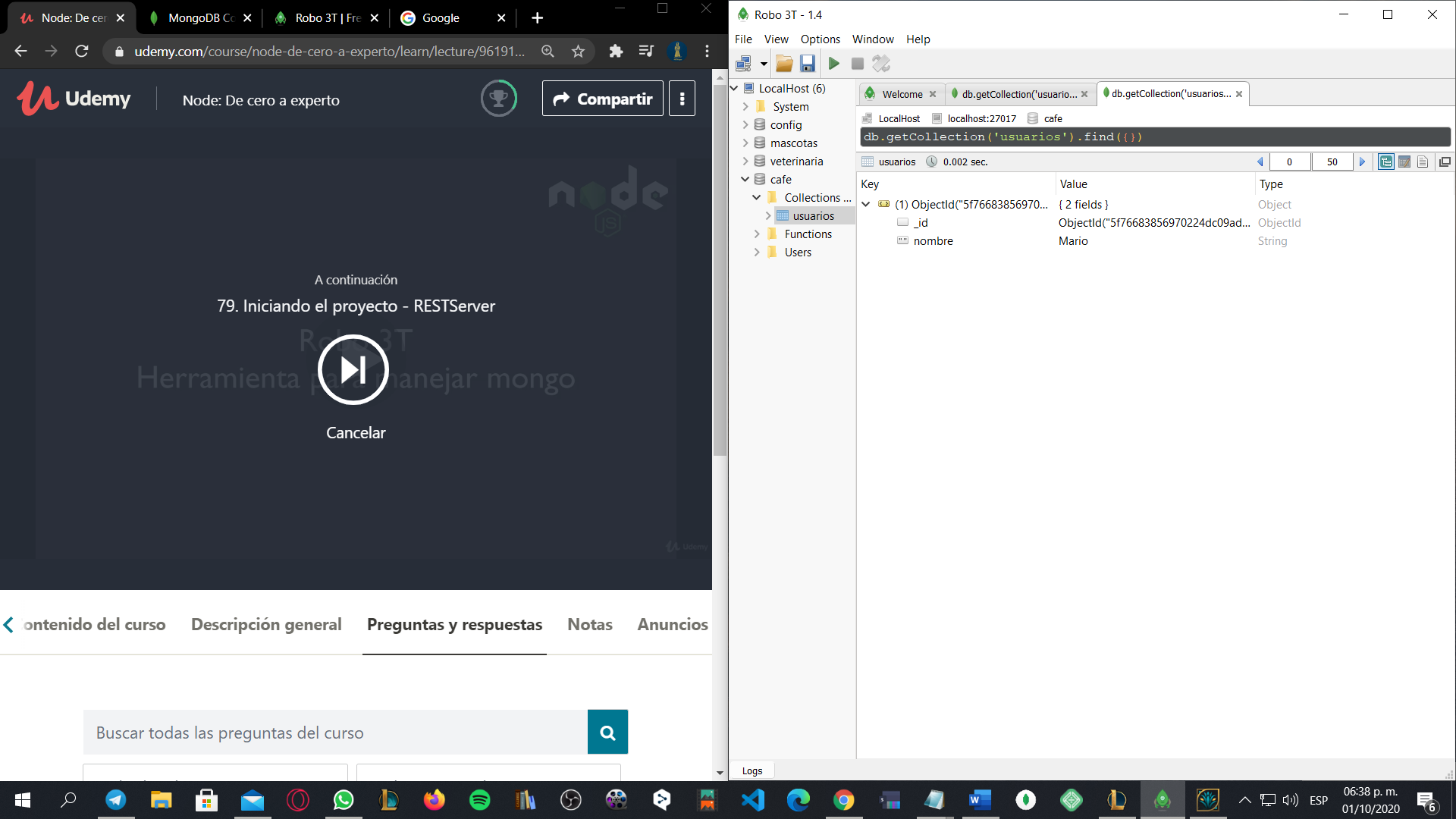


Creamos una colección y click izquiero podemos insertar los documentos



Ya adentro el jason ejemplo

“nombre”:”Mario”



**INICIANDO PROYECTO REST SERVER**

Npm init

Trabajaremos con express

npm install express –save

creamos capreta nueva server con el archivo serve.js

para ejecutarlo node server/server por que esta dentro de la carpeta

const express = require('express')

const app = express()

app.get('/', function(req, res) {

    res.send('Hello World')

})

app.listen(3000, () => {

    console.log('Escuchando el puerto 3000');

})

Como vamos a trabajar con Json el método send lo cambiamos por json

Mejor usar nodemon server/server para que este atento a los cambios

**PETICIONES HTTP - GET -PUT -POST -DELETE**

Vamos a ahacer las 4 peticiones básicas que necesitaremos

const express = require('express')

const app = express()

app.get('/usuario', function(req, res) {

    res.json('get usuario')

})

app.post('/usuario', function(req, res) {

    res.json('post usuario')

})

app.put('/usuario', function(req, res) {

    res.json('put usuario')

})

app.delete('/usuario', function(req, res) {

    res.json('delete usuario')

})

app.listen(3000, () => {

    console.log('Escuchando el puerto 3000');

})

Como nosotros vamos a modificar un usuario en especifico necesitamos poner el comid para que acepte cualquier id, y agregamos la variable id que sale del URL

app.put('/usuario/:id', function(req, res) {

    let id= req.params.id;

    res.json({

        id

    })

})

Trabajamos con el post, tenemos que obtener las variables que mande el cliente..Existe un paquete que se llama body-parser

npm install body-parser –sabe

const bodyParser = require('body-parser');

e implemantamos los middleware

const bodyParser = require('body-parser');

// parse application/x-www-form-urlencoded

app.use(bodyParser.urlencoded({ extended: false }))

// parse application/json

app.use(bodyParser.json())

con una variable que es la que,va a tiener toda la información que necesitamos, las peticviones se guardaran en esa variable

app.post('/usuario', function(req, res) {

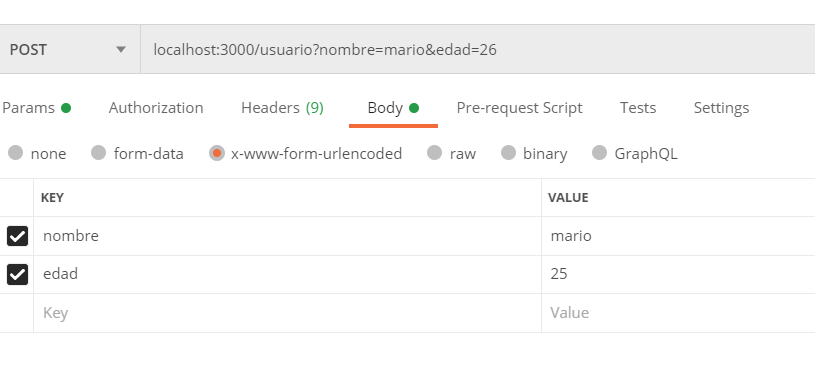
    let body = req.body;

    res.json({

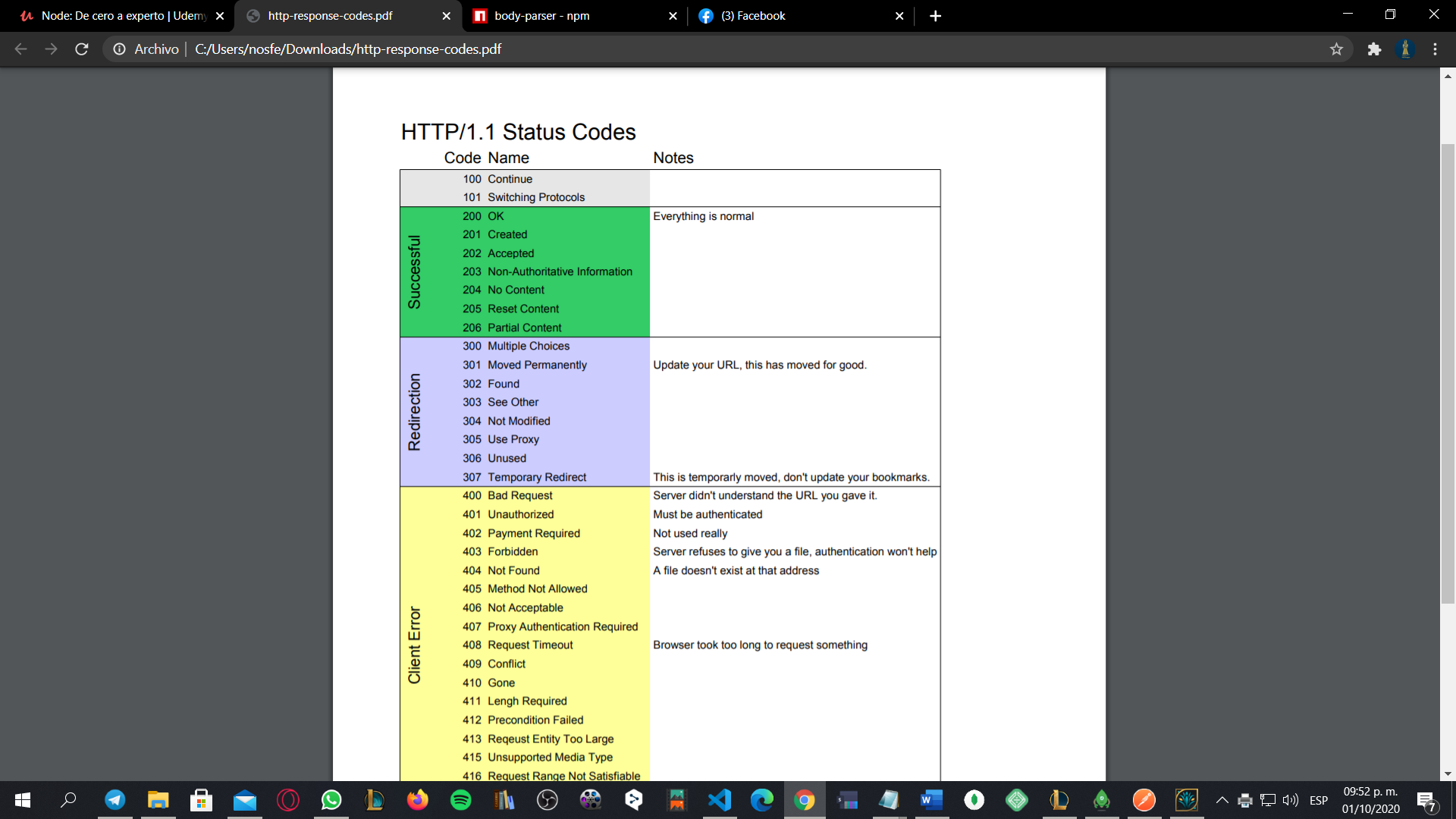
        body

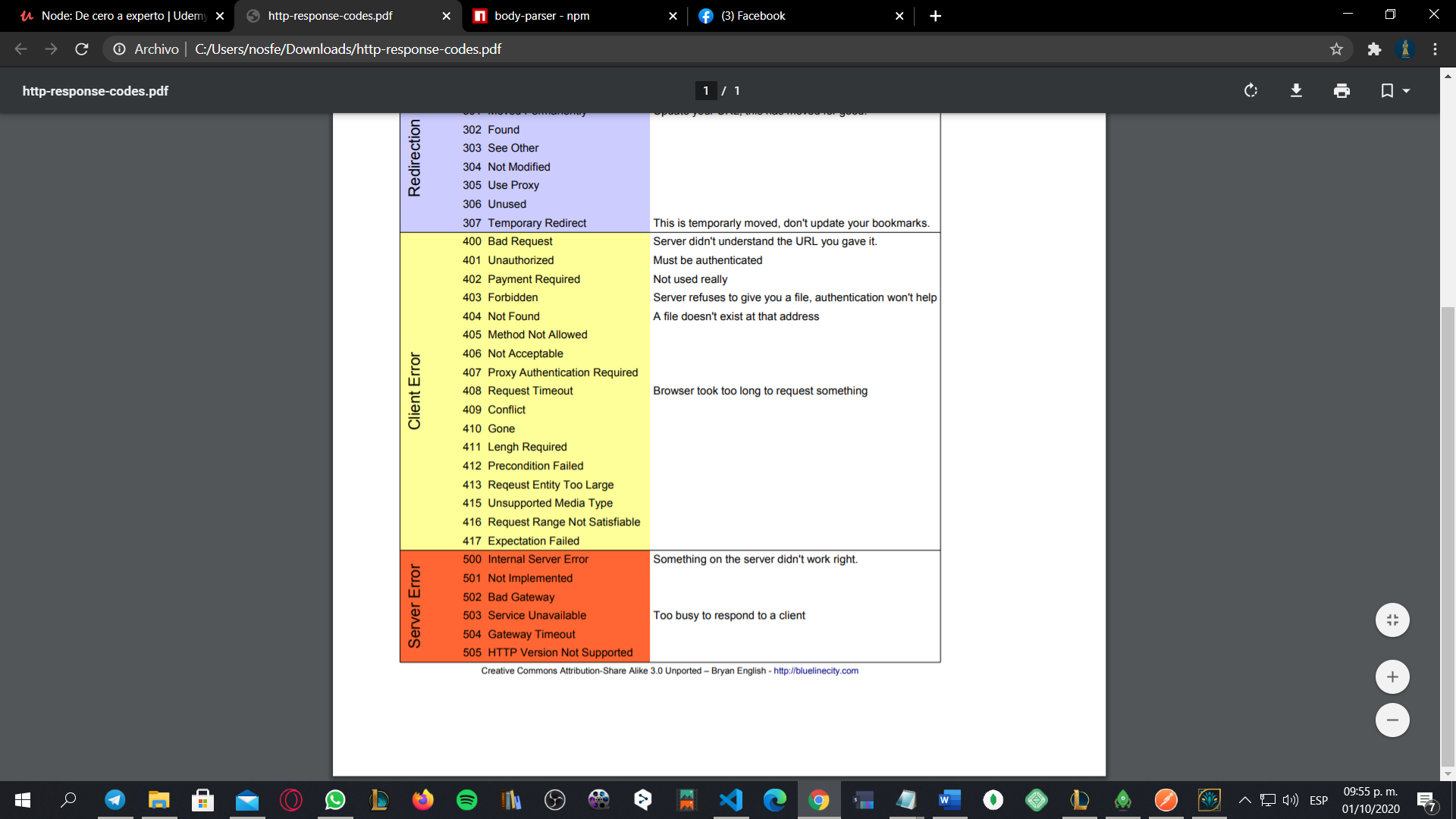
    })

Ya salió la petición pero por x-www-form-urlenconde



**CODIGOS DE RESPUESTAS HTTP**





Quedaría un post con el estatus 400

app.post('/usuario', function(req, res) {

    let body = req.body;

    //estatus 400 Bad request

    res.status(400).json({

        ok: false,

        mensaje: 'El nombre es necesario'

    })

    res.json({

        persona: body

    })

})