//creacion de esquema para cada documento (conjunto de datos)

const userSchema = new mongoose.Schema ({ //esquema creado por medio de la clase "mongoose.Schema"

    email: String,

    password: String

});

const secret = "Thisisourlittlesecret.";

userSchema.plugin(encrypt, { secret: secret, encryptedFields: ["password" ] }); //Esto se debe hacer antes de crear el modelo (siguiente linea) el segundo parametro se usa para indicar qué elemento vamos a encriptar

//creacion de modelo de tabla con base en el esquema (la cual mongoDB pluraliza posteriormente)

const User = new mongoose.model("User", userSchema);

app.get("/", function(req, res) {

    res.render("home");

})

//VENTANA LOGIN

app.get("/login", function(req, res) {

    res.render("login");

});

app.post("/login", function(req, res){

    const username = req.body.username;

    const password = req.body.password;

    //verificamos si el email del usuario se encuentra registrado, en caso tal se verifica si la contraseña ingresada es la misma que se encuentra almacenada

    User.findOne({email: username}, function(err, foundUser) {

        if(err) {

            console.log(err);

        } else {

            if(foundUser) {

                if(foundUser.password === password) {

                    res.render("Secrets");

                }

            }

        }

    });

});

//VENTANA REGISTER

app.get("/register", function(req, res) {

    res.render("register");

});

app.post("/register", function(req, res) {

    const newUser = new User({

        email: req.body.username, //se relaciona con el input que se encuentra en register.ejs cuyo "name" es "username"

        password: req.body.password  //se relaciona con el input que se encuentra en register.ejs cuyo "name" es "password"

    });

    newUser.save(function(err) { //justo en este momento, al ejecutar .save, se realiza la encriptacion del dato seleccionado previamente, en ete caso se encripta "password"

        if(err) {

            console.log(err);

        } else {

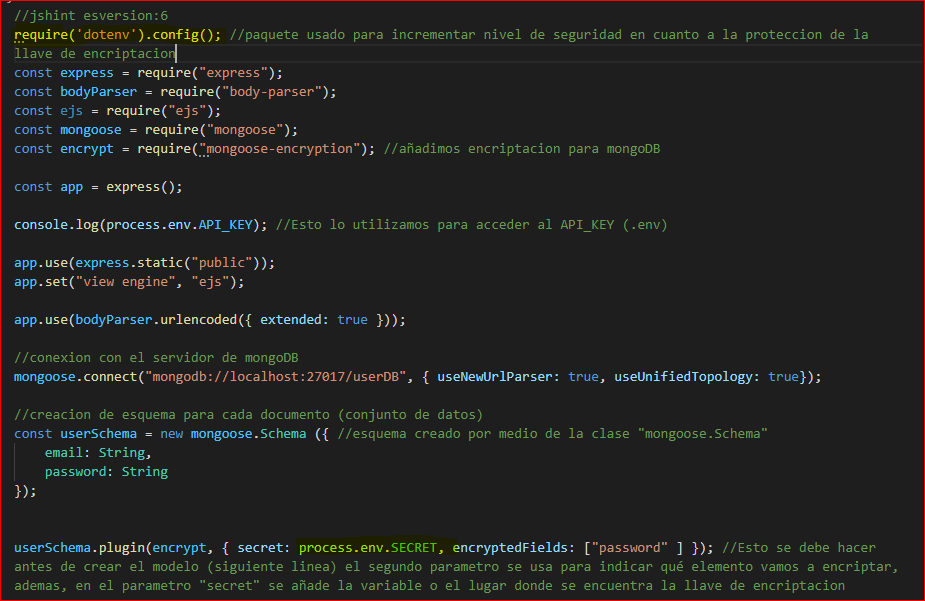
            res.render("secrets");

        }

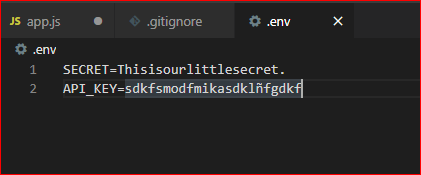
    });

});

SEGUNDO NIVEL DE ENCRIPTACION



PROTECCION DE LA LLAVE DE ENCRIPTACION



ARCHIVO DONDE SE GUARDA LA LLAVE “SECRET”