**Github**

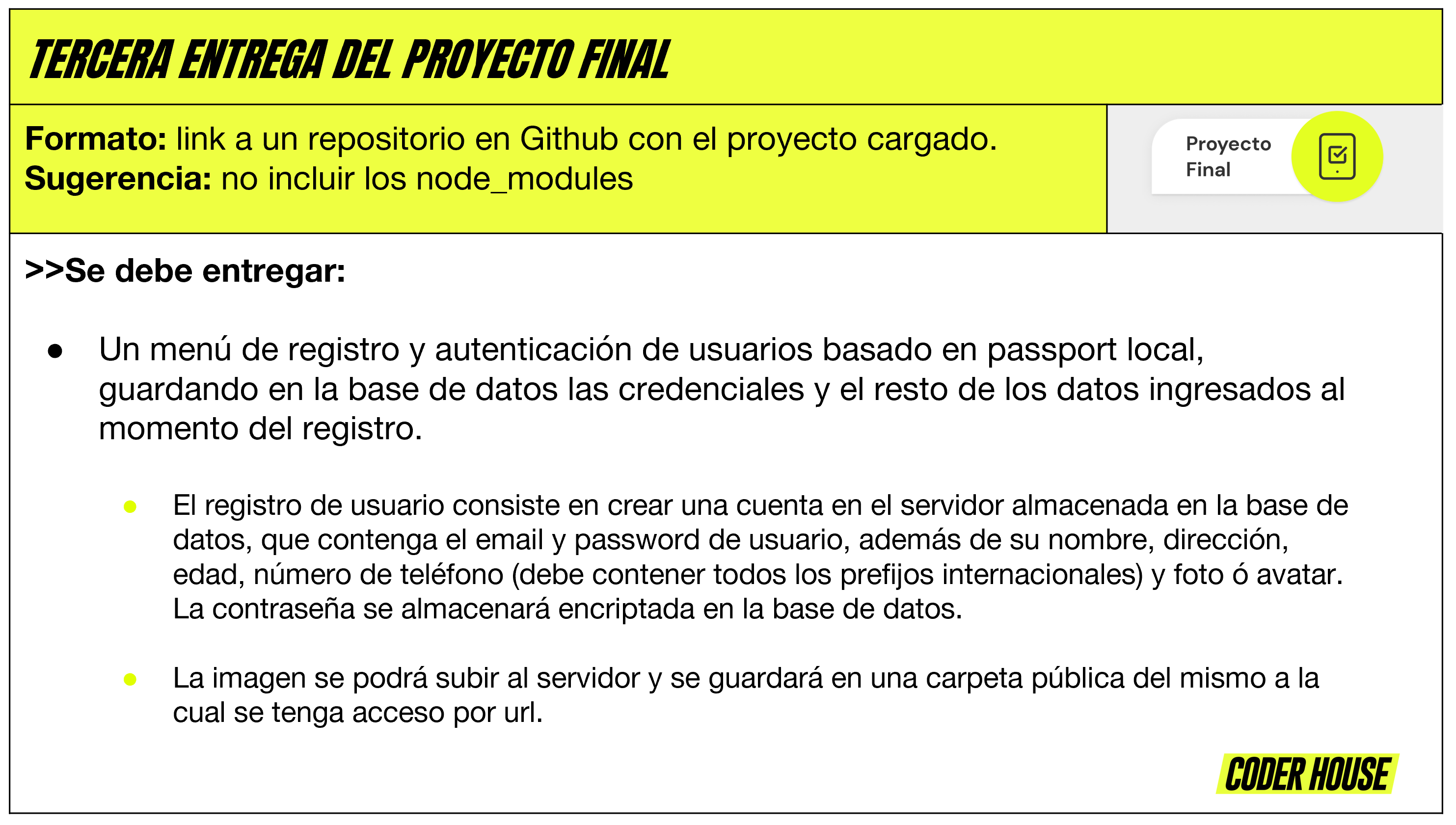
* <https://github.com/nancybarea/nodeJS_TPFinal_backend>

**heroku**

* <https://ecommerce-ba.herokuapp.com/>
* <https://ecommerce-ba.herokuapp.com/registrarse>
* <https://ecommerce-ba.herokuapp.com/login>
* <https://ecommerce-ba.herokuapp.com/abmUsuarios>
* <https://ecommerce-ba.herokuapp.com/abmProductos>
* <https://ecommerce-ba.herokuapp.com/abmPedidos>
* <https://ecommerce-ba.herokuapp.com/abmCarritos>
* <https://ecommerce-ba.herokuapp.com/abmMensajes>
* <https://ecommerce-ba.herokuapp.com/subirArchivos>
* <https://ecommerce-ba.herokuapp.com/failLogin>

A continuación algunas notas sobre lo pedido, como los resolvi, pantallas y demás

(si falta algo que no nombre, se puede ver en github o heroku)



**Parte del código 🡪 usuarios.js** (para mostrar que uso Passport)

//estragegia de registro

passport.use('registro', new Strategy({passReqToCallback:true},async (req, username, password, done)=>{

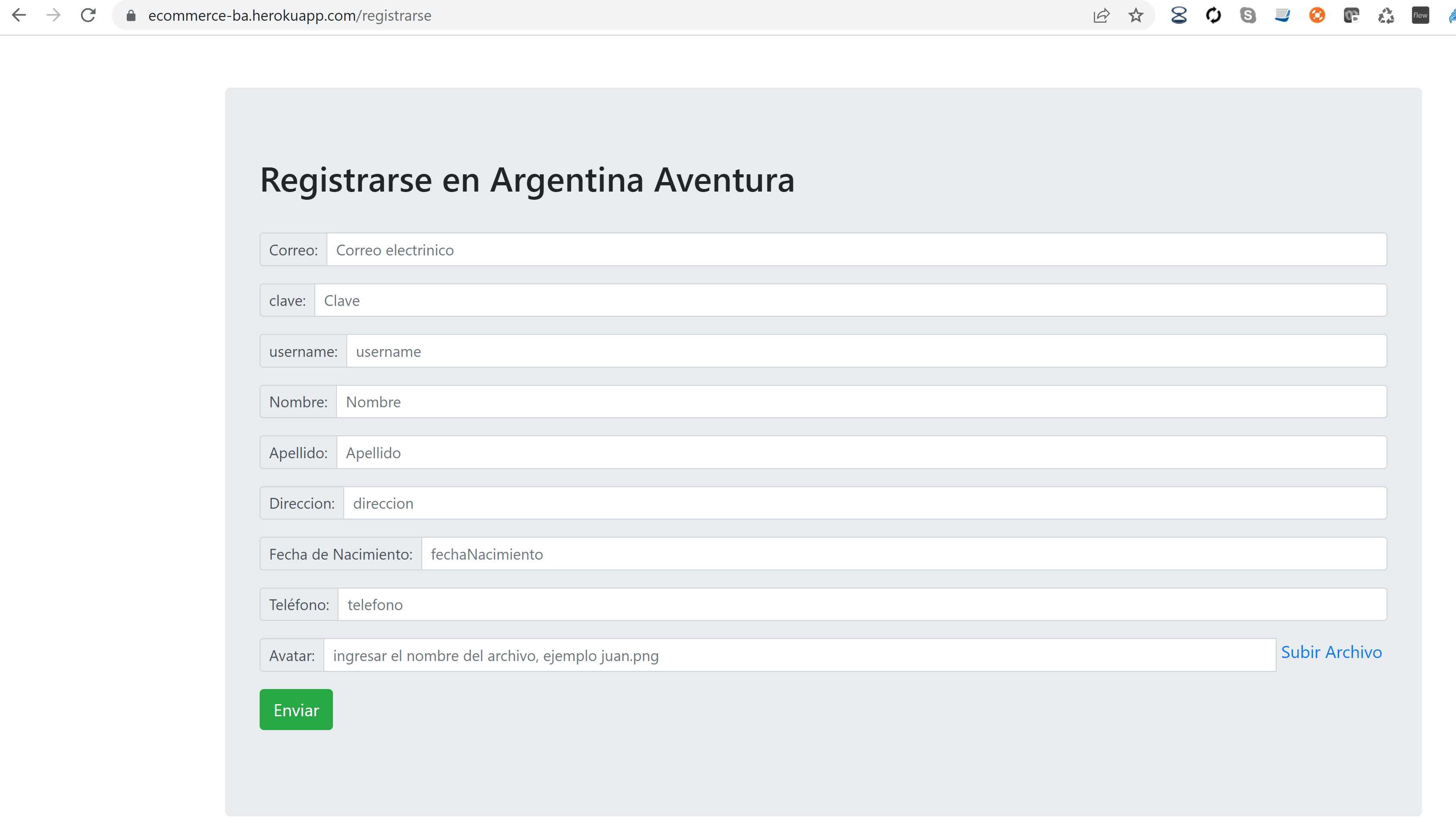
    logger.info(`usuarios.js - passport registro`)

**Registro API desde postman**



**Registro API desde web (con plantillas EJS)**

[**https://ecommerce-ba.herokuapp.com/registrarse**](https://ecommerce-ba.herokuapp.com/registrarse)



**Como guarda en la base de datos**



**Se observa que la contraseña esta encriptada use bcrypt**

function createHash(password) {

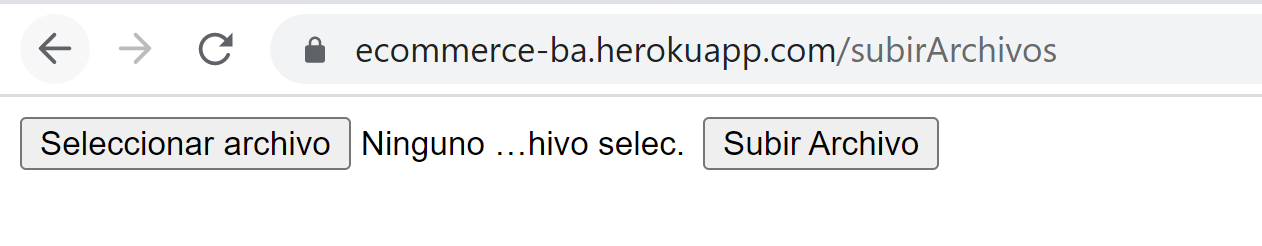
    return bCrypt.hashSync(password, bCrypt.genSaltSync(10), null);

}

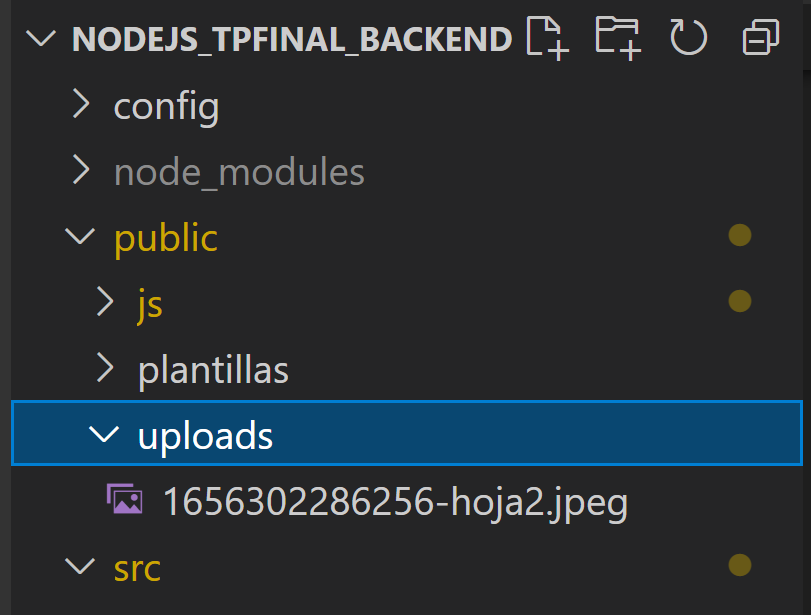
**Subir archivos -- use Multer**

**Hice la url , que es donde te envia desde el formulario de registro actualmente ( a futuro quizá lo haga diferente)**

[**https://ecommerce-ba.herokuapp.com/subirArchivos**](https://ecommerce-ba.herokuapp.com/subirArchivos)



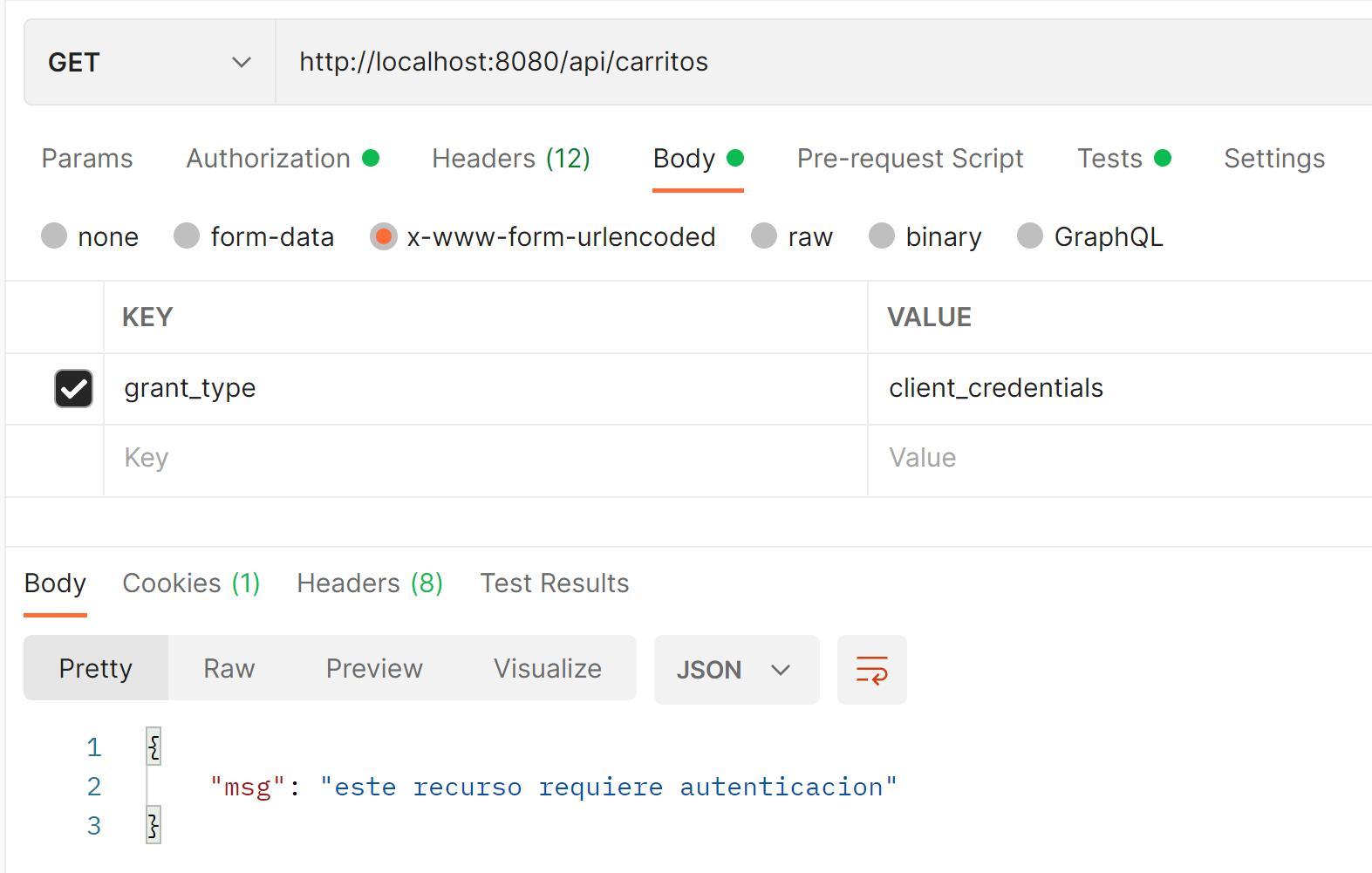
**Y lo guarda en la carpeta uploads de public**



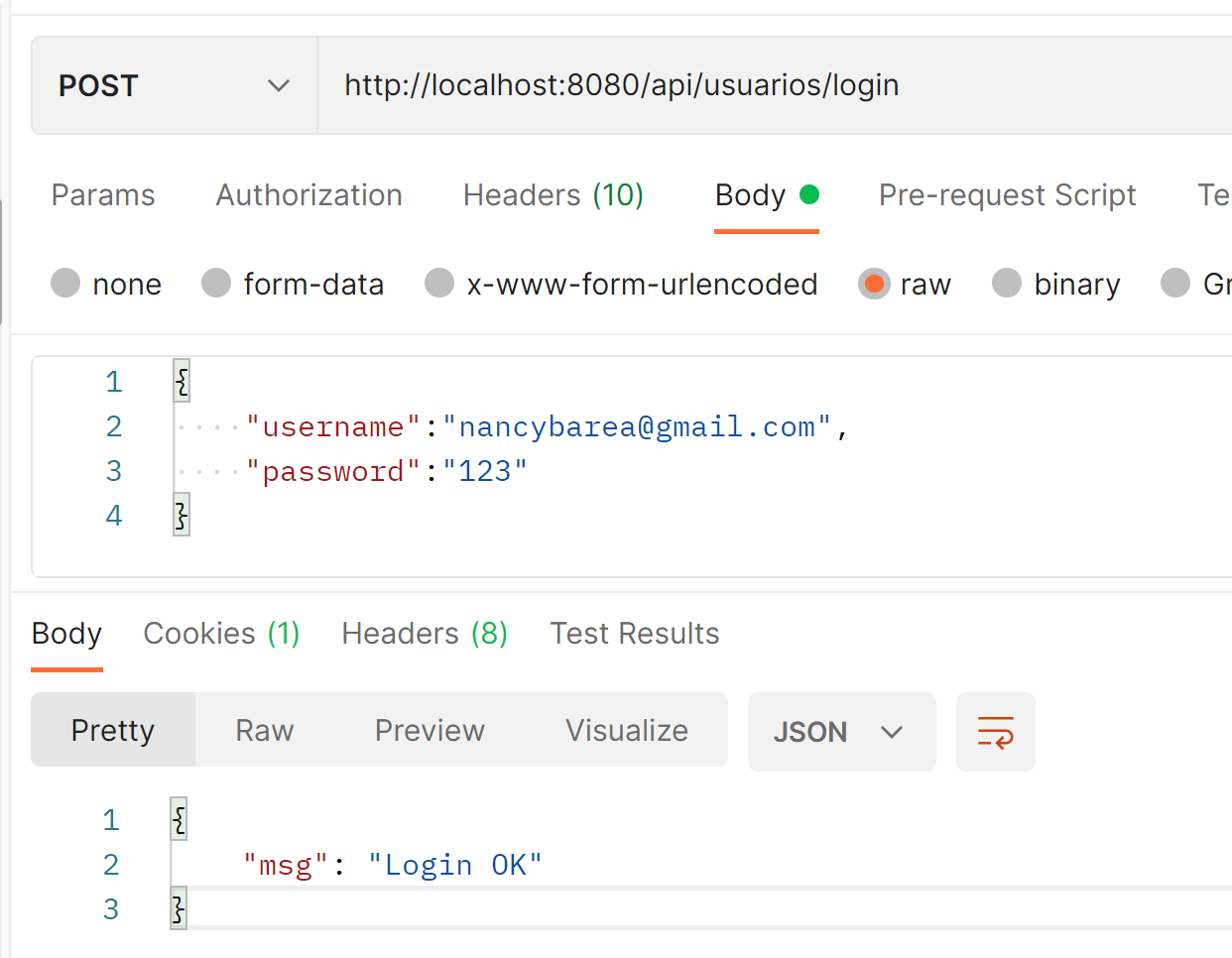
**Se puede observar como funciona autenticación**

En el código 🡪 CarritosRoutes.get('/', requiereAutenticacion, carritosController.obtenerCarritos)

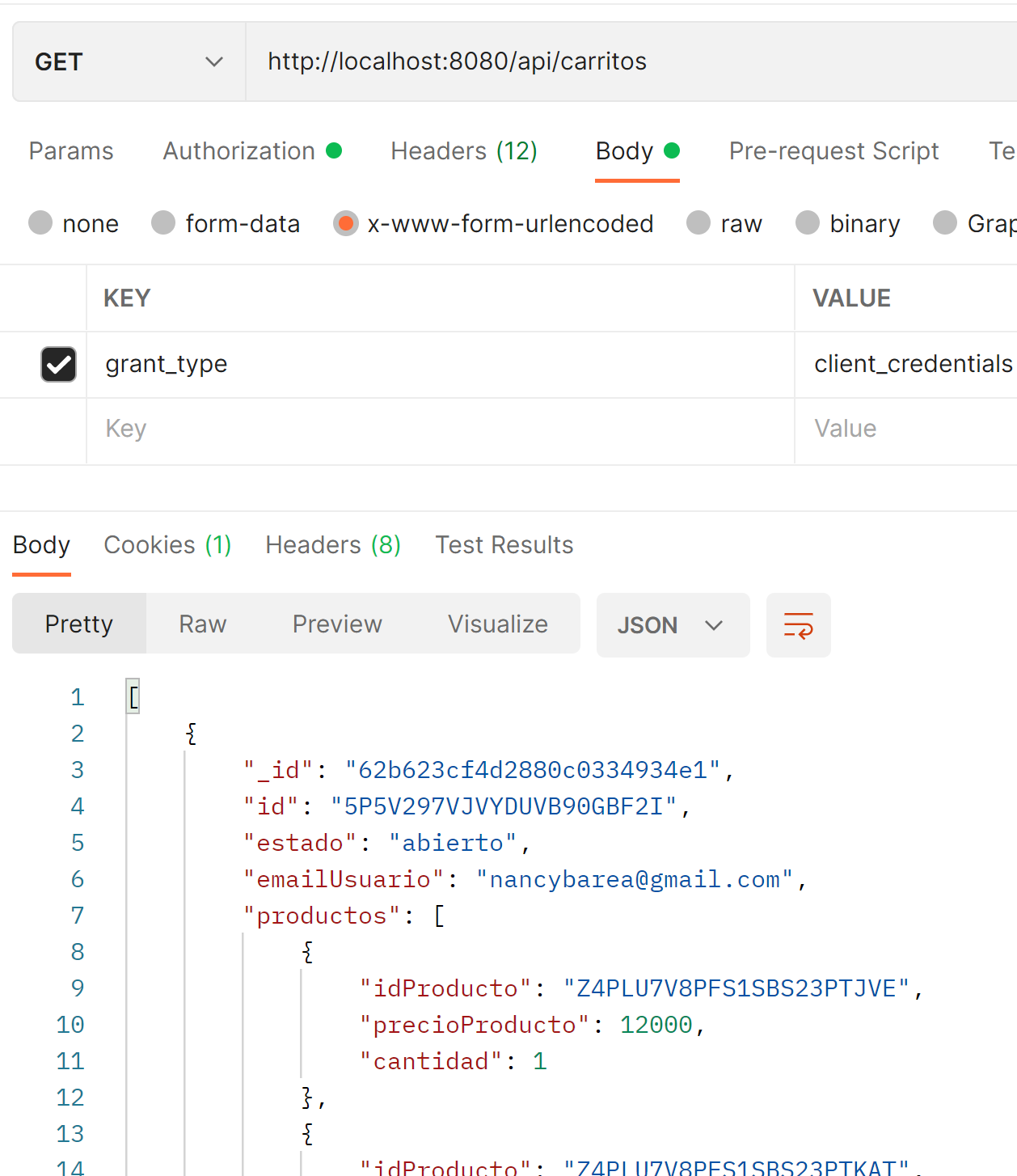
* Cuando pruebo muestra error autenticación

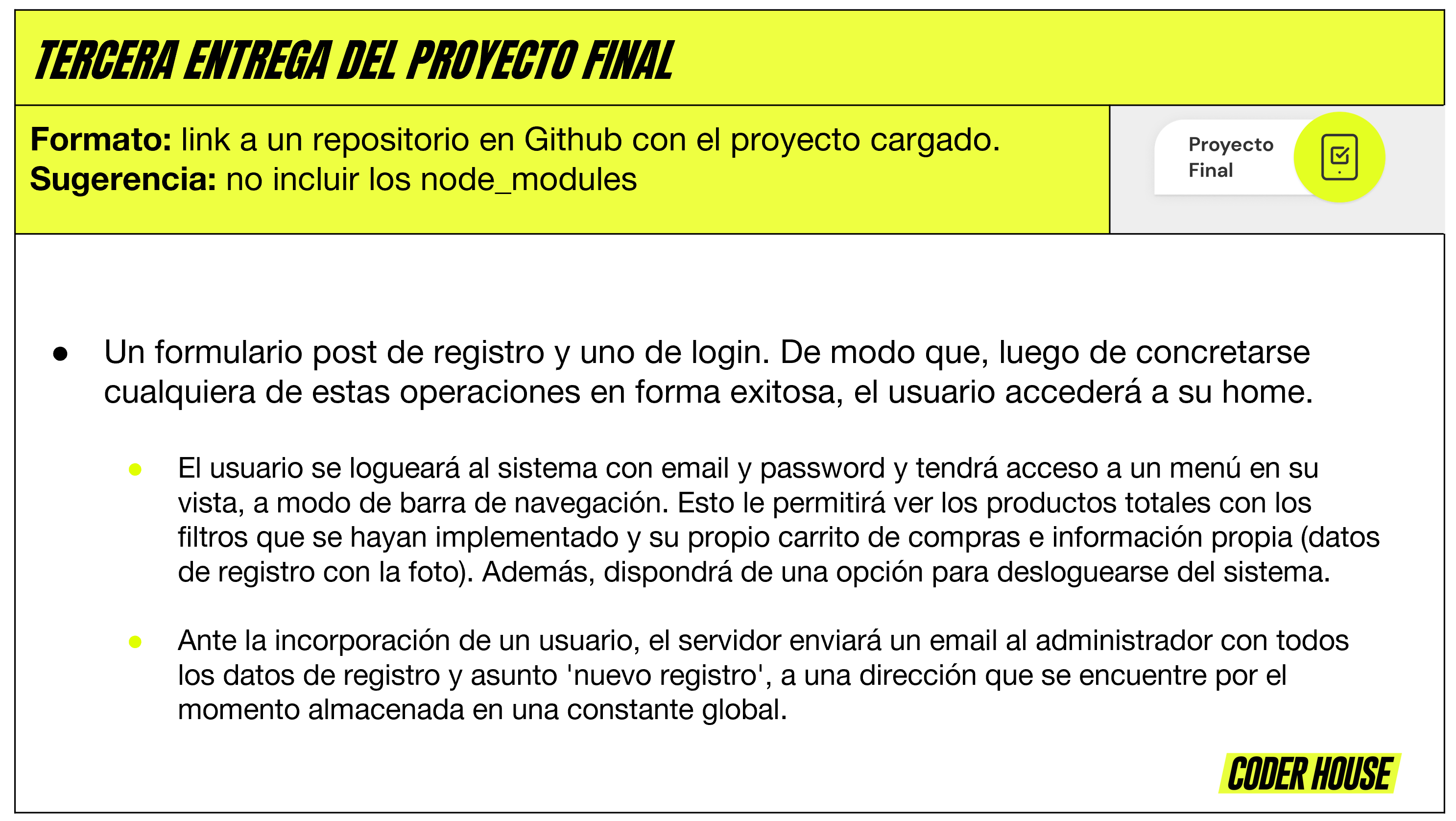


Me logueo …



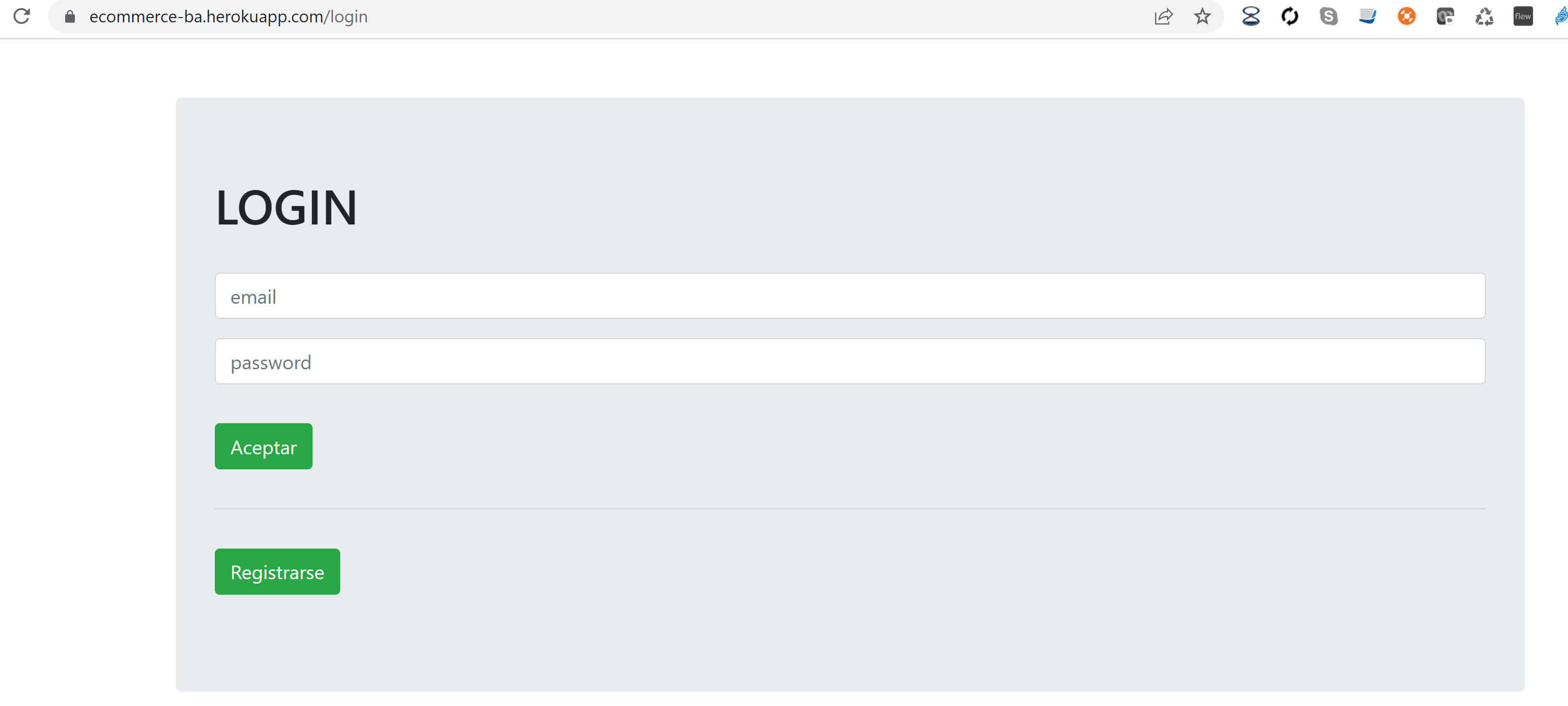
Y ahora si puedo ver carritos…



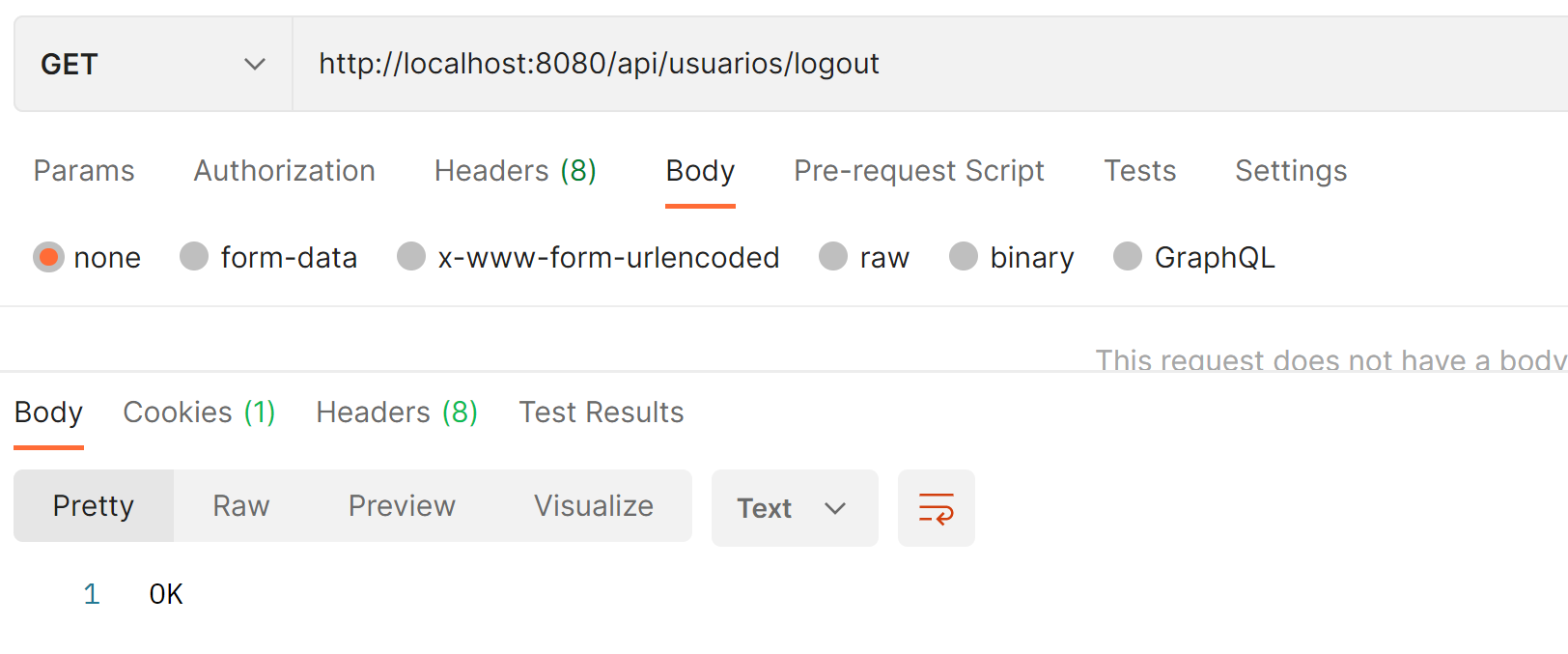


**El login y registración se mostró antes, agrego**

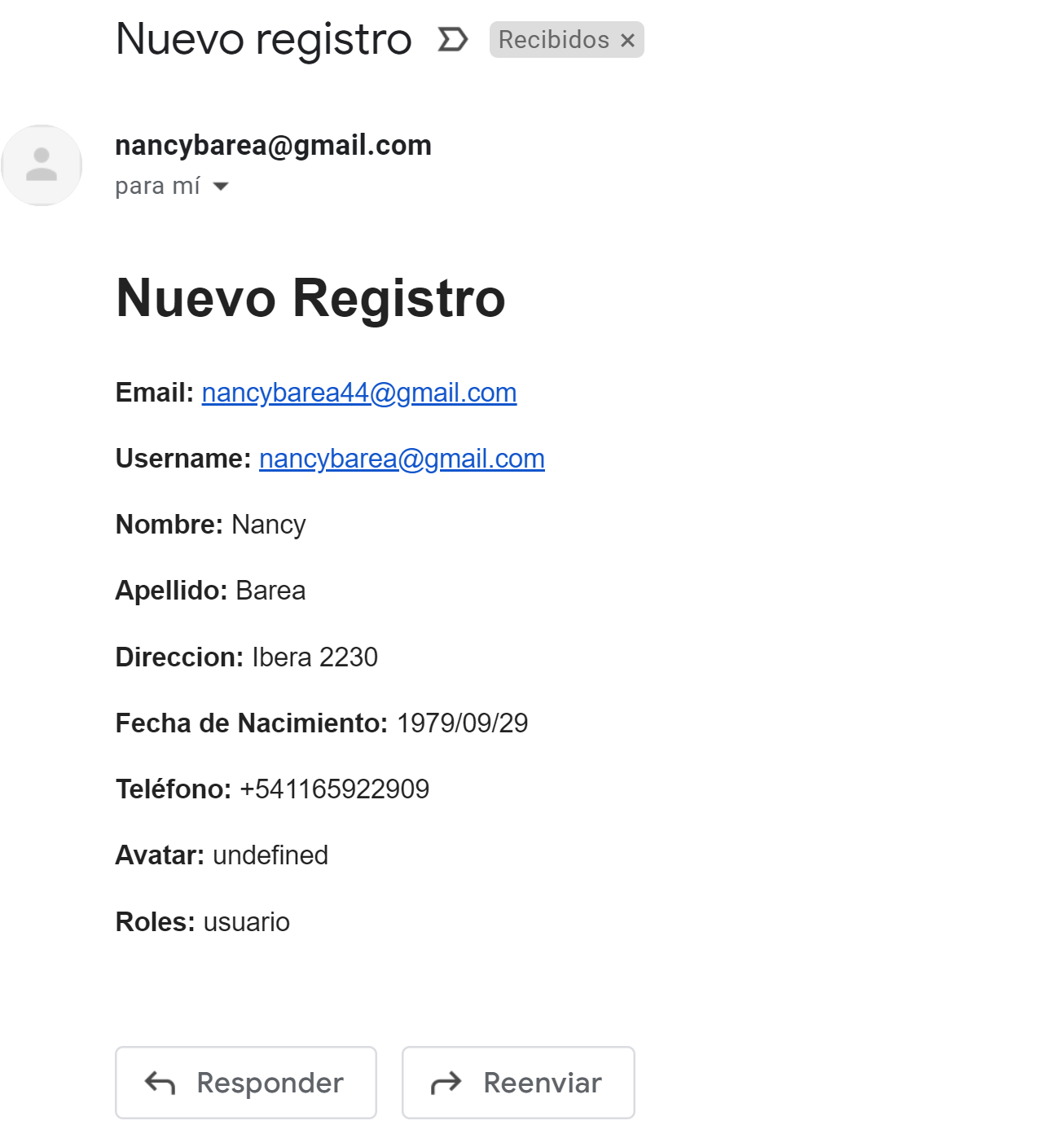
**Pantalla web de login**

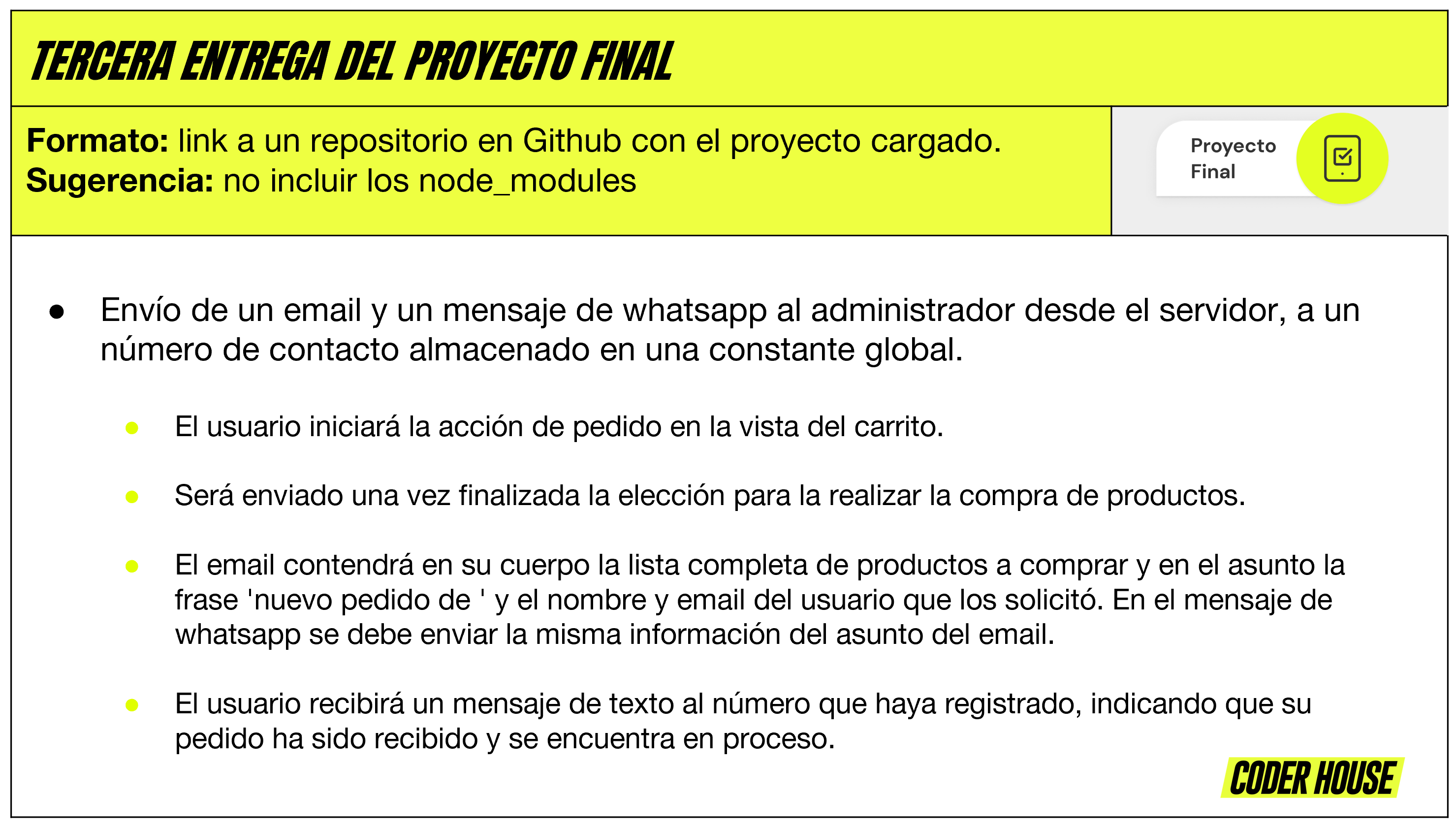


**Y la api de deslogue**



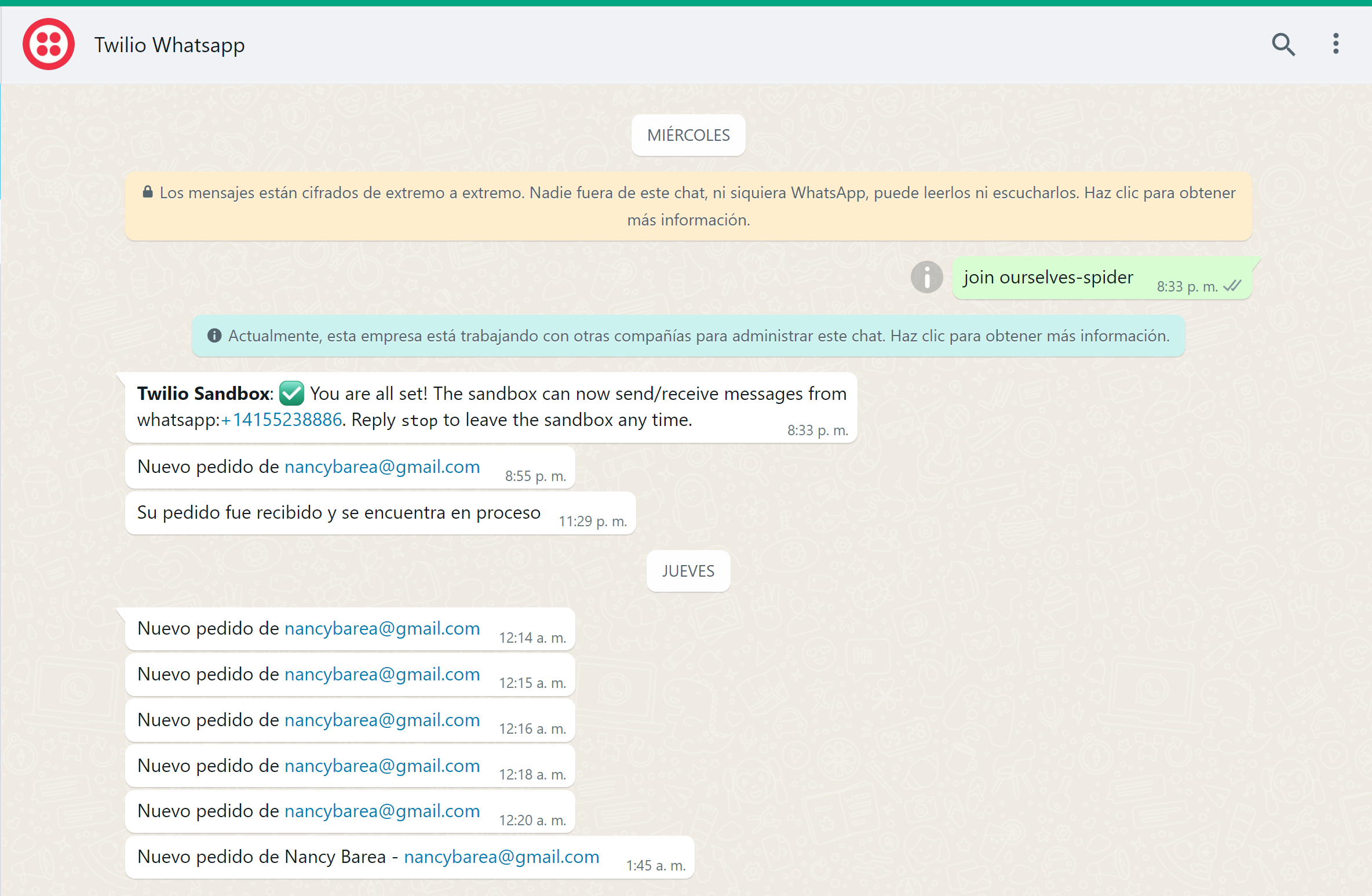
**Envio de email, lo hice con mi gmail**





**Envio whatsapp**

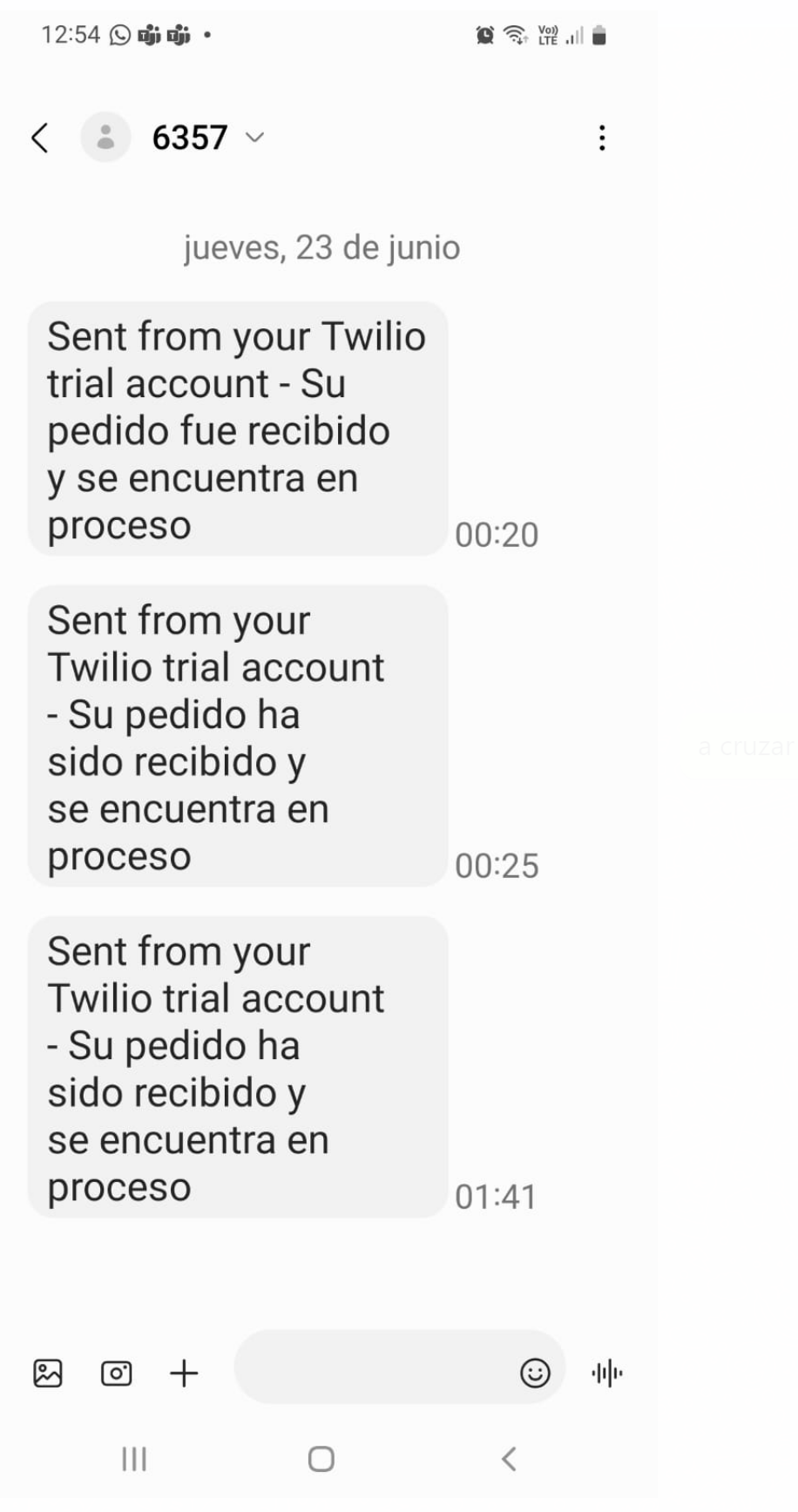
Paso todas las pruebas que estuve haciendo, la ultima se puede ver como quedo



**Email con el mismo asunto que whatsapp**



**SMS que envio a usuario, es mi número de celular, desde Twilio, para avisar que se hizo la compra**



**En cuanto al código cree una clase PEDIDOS para poder manejar las compras , es decir una vez que confirma el carrito, que dice que quiere comprar, pasaría a PEDIDOS y ahí se comienza todo el proceso de compra**

    async addPedido(objeto) {

        try{

            //cargo el pedido

            const pedido = new PedidosDto(objeto)

            await this.pedidosDao.add(pedido);

            logger.info(`Registro de pedido Ok `);

            //obtengo los datos del usuario

            const usuario = await this.usuariosDao.getByEmail(pedido.email)

            //envio de notificaciones al admin y usuario

            await this.enviarEmailNuevoPedido(pedido, usuario.nombre, usuario.apellido)

            await this.enviarWhatsappNuevoPedido(pedido.email, usuario.nombre, usuario.apellido)

            await this.enviarSMSPedidoEnProceso(usuario.telefono)

            return pedido.get();

        }

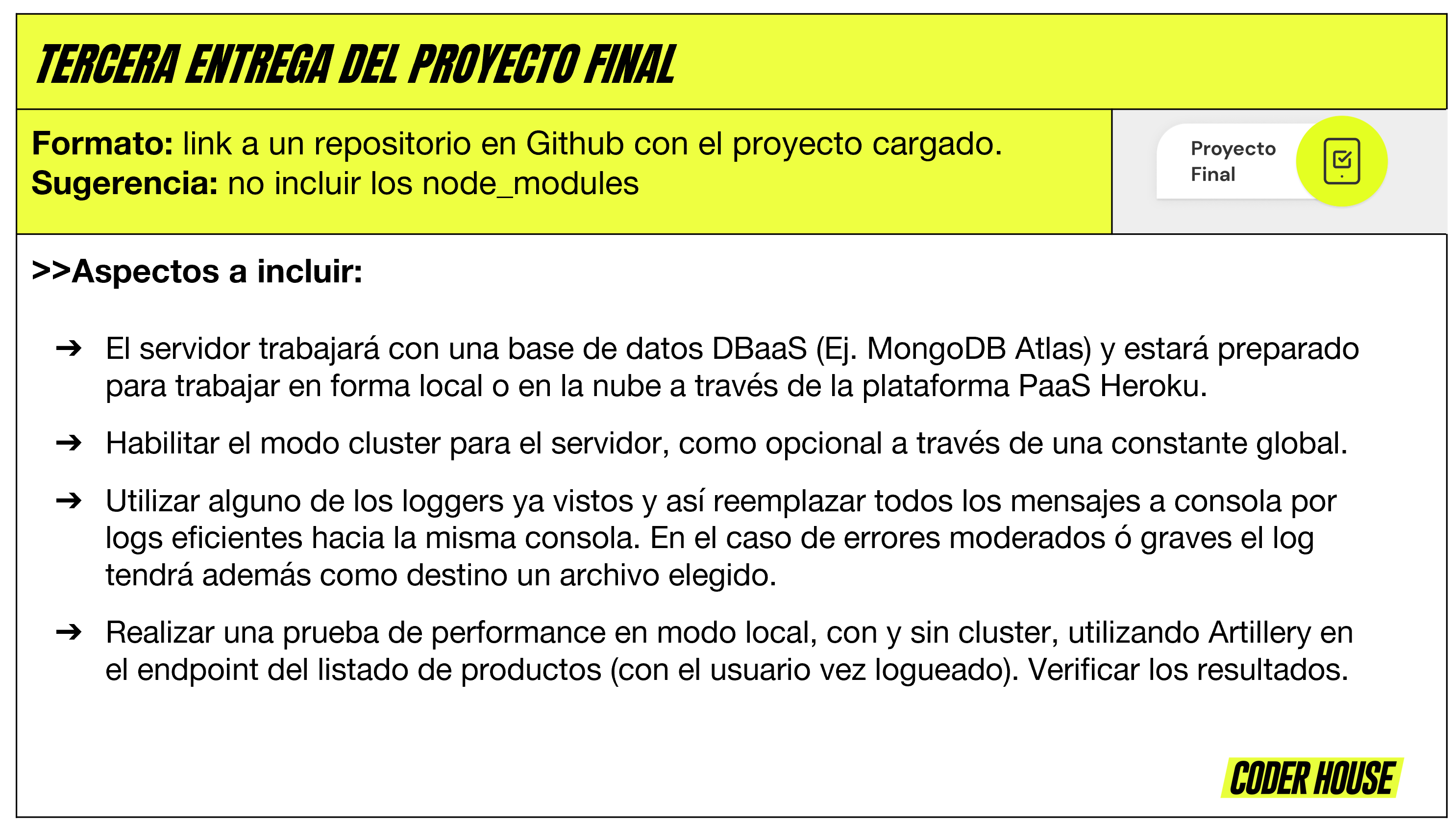
        catch (err){

            logger.error(`Error al agregar un pedido: ${err}`);

            throw new CustomError(401, `Error al agregar un pedido`, err)

        }

    }



**Todos estos temas se mencionaron anteriormente y se pueden ver en el código**

**Agrego lo de cluster (que fue presentado en desafíos anteriores)**

 "scripts": {

    "start": "node .",

    "start:cluster": "node src/app.js --modo CLUSTER",

**Nota: estaré mejorando esta entrega ya que va ser la del trabajo practico final. Las apis se encuentran funcionando todas OK, queda terminar el frontend para poder utilizar TODAS las apis que ya están listas, por ahora solo aplique algunas pero ire mejorando en estos días hasta la entrega final el proyecto.**