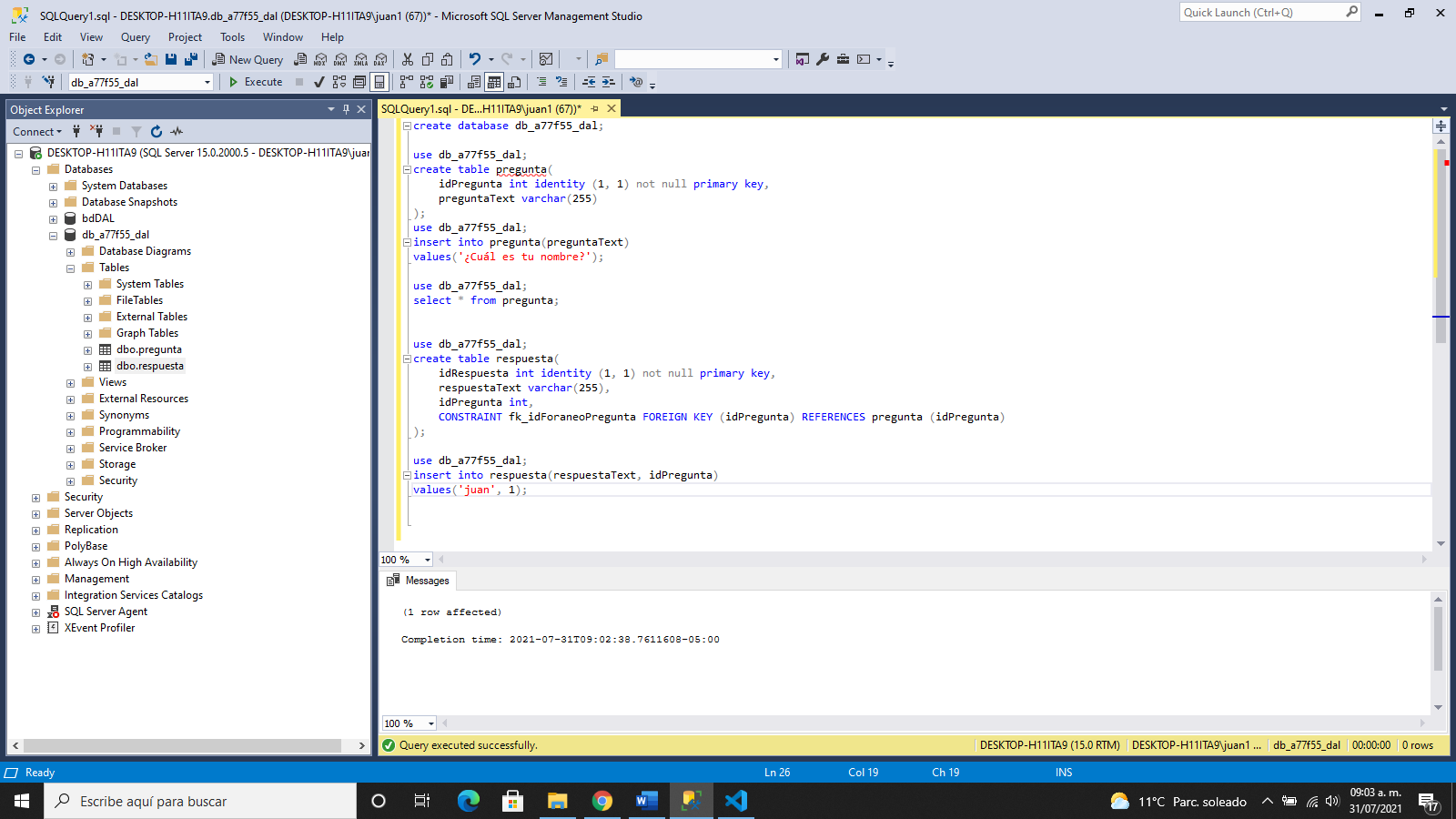
1. Creamos una base de datos en SQL SERVER con el administrador Management Studio 18



create database db\_a77f55\_dal;

use db\_a77f55\_dal;

create table pregunta(

idPregunta int identity (1, 1) not null primary key,

preguntaText varchar(255)

);

use db\_a77f55\_dal;

insert into pregunta(preguntaText)

values('¿Cuál es tu nombre?');

use db\_a77f55\_dal;

select \* from pregunta;

use db\_a77f55\_dal;

create table respuesta(

idRespuesta int identity (1, 1) not null primary key,

respuestaText varchar(255),

idPregunta int,

CONSTRAINT fk\_idForaneoPregunta FOREIGN KEY (idPregunta) REFERENCES pregunta (idPregunta)

);

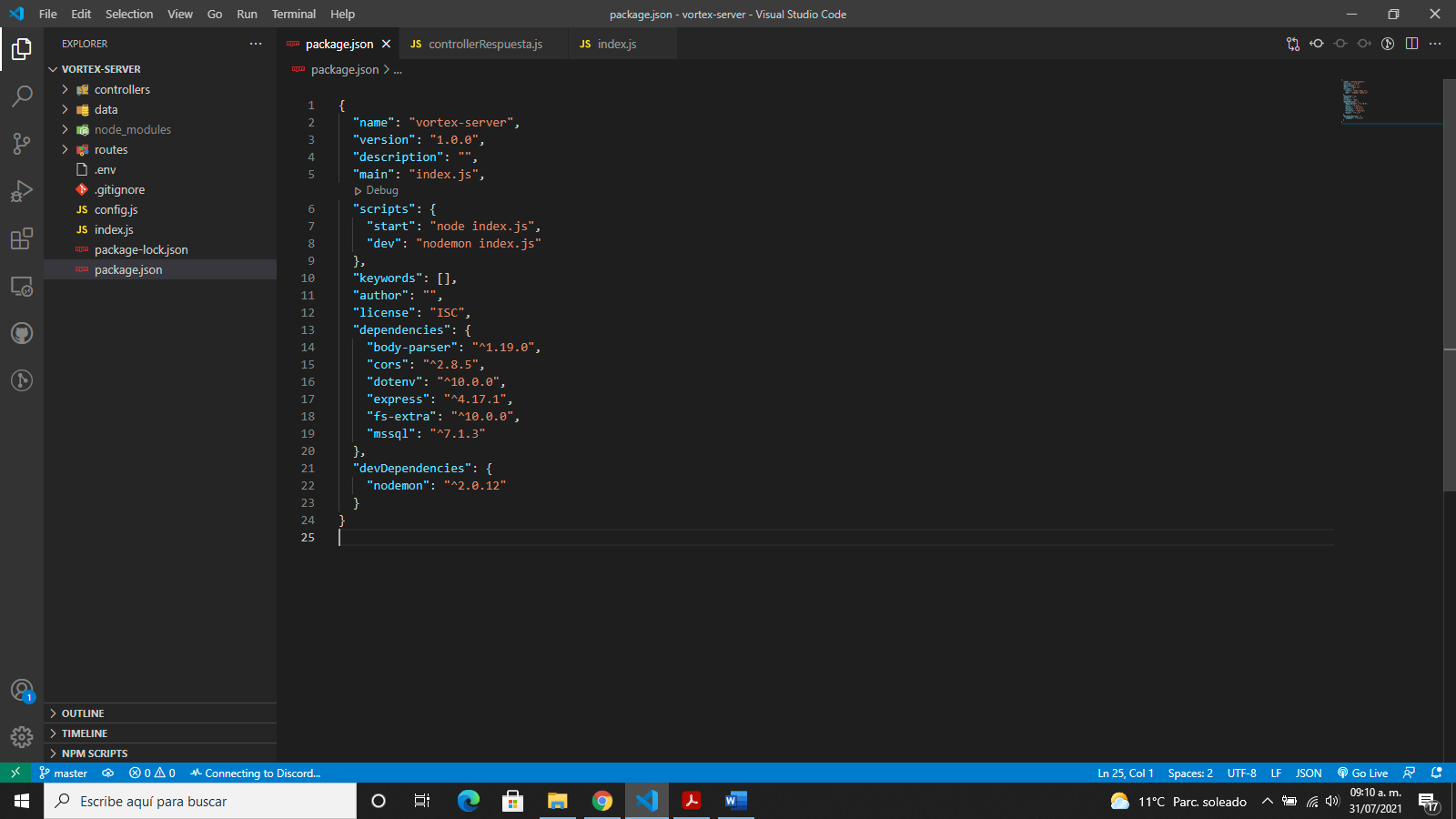
use db\_a77f55\_dal;

insert into respuesta(respuestaText, idPregunta)

values('juan', 1);

1. Creamos un proyecto en Node.js con nuestro editor de texto Visual Studio Code y instalamos las dependencias que nos ayudaran a crear un servidor web y una conexión con SQL SERVER.

* npm start -y
* npm install express mssql fs-extra cors body-parser dotenv

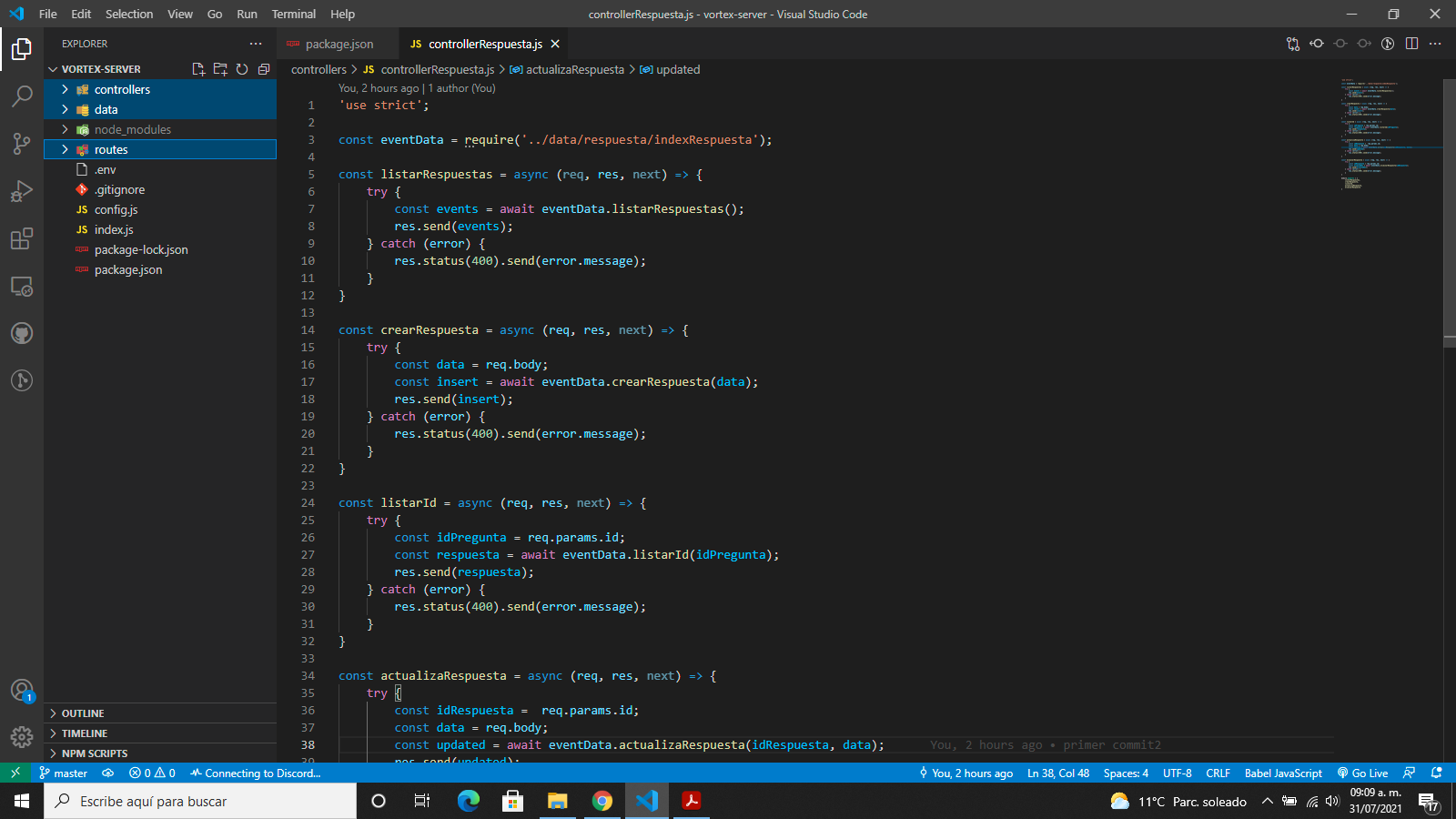


1. Creamos la estructura del SERVIDOR WEB

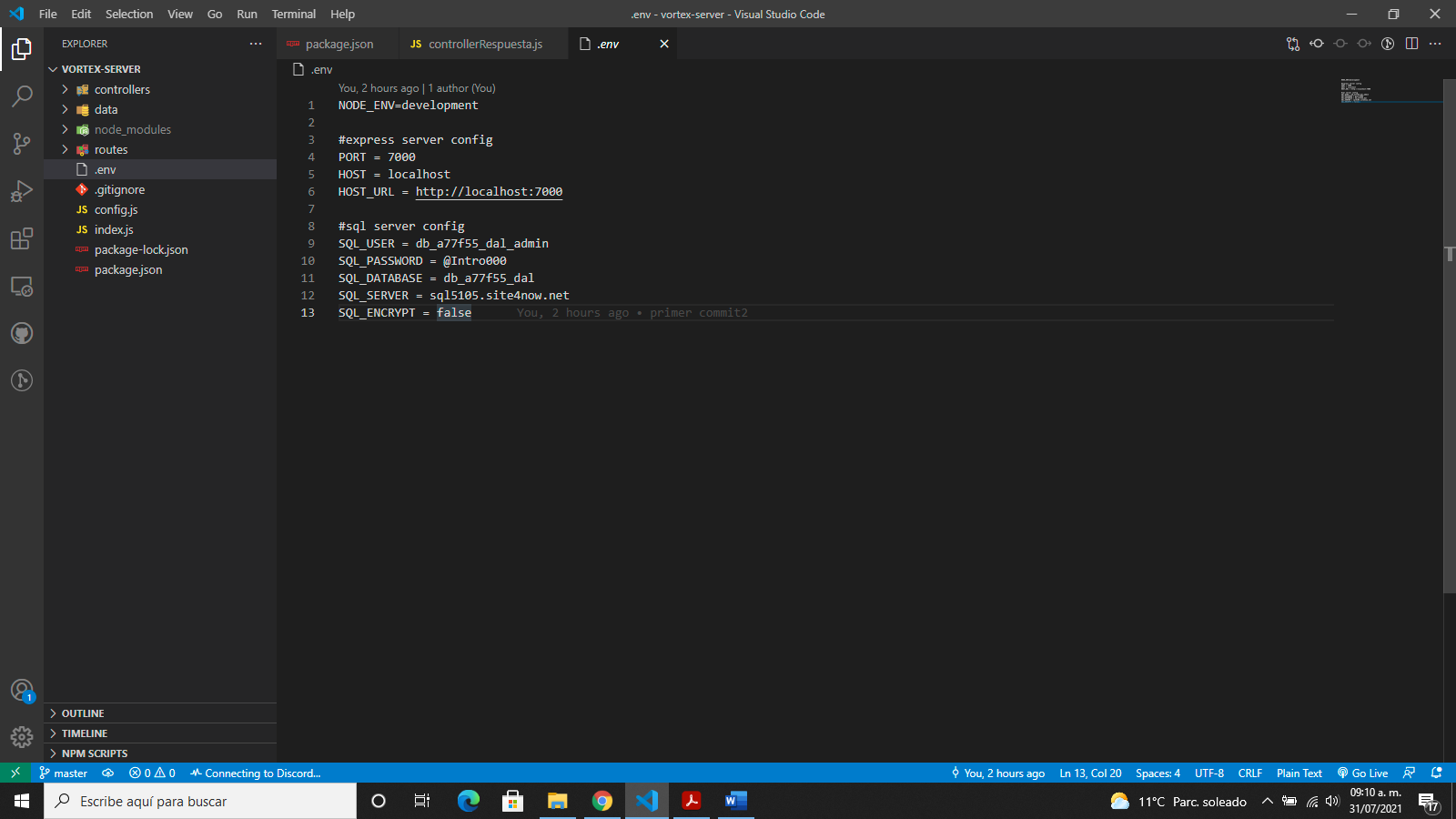
-Data

-Controller

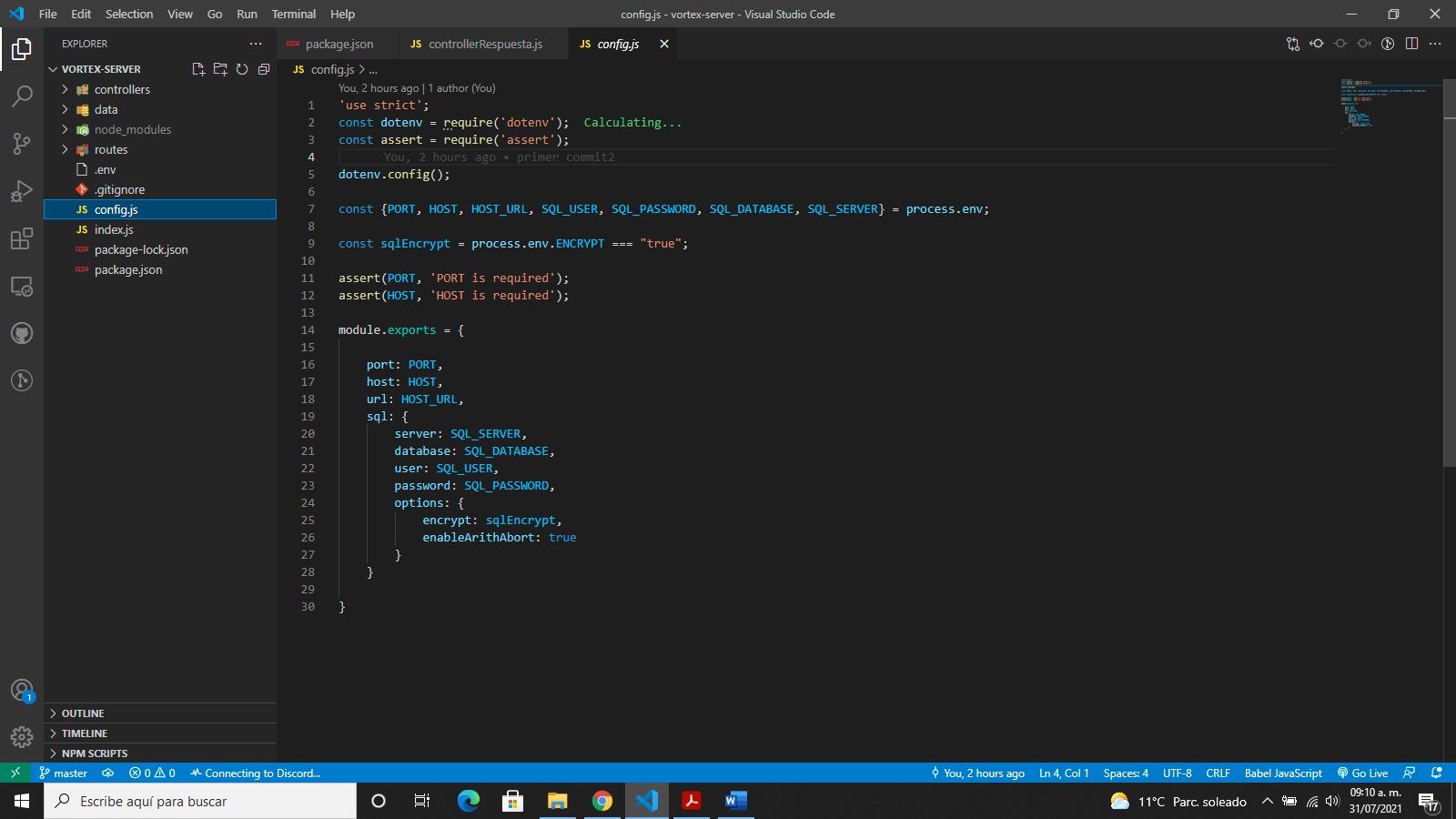
-Routes



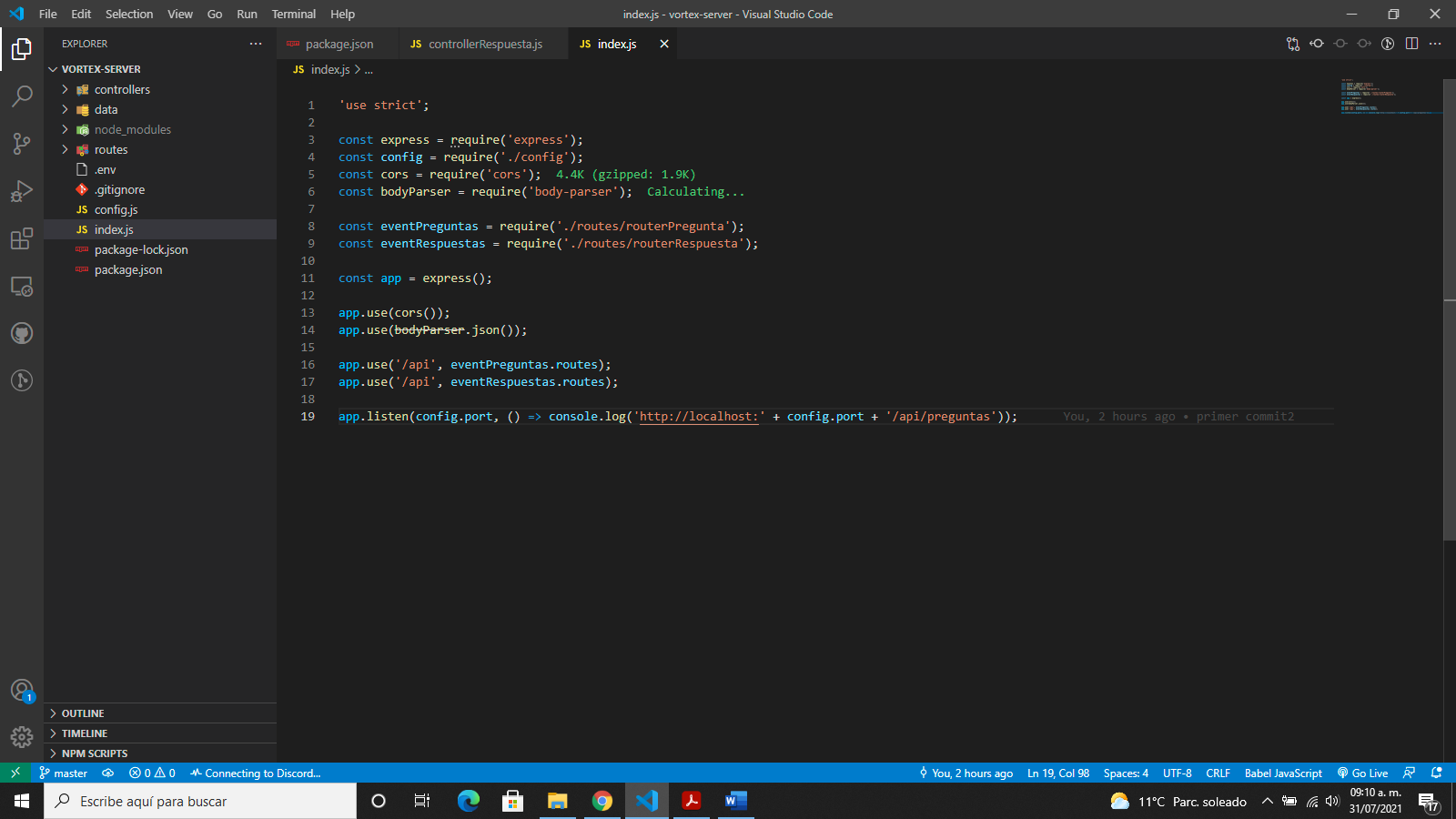
1. Creamos el archivo .env para configurar nuestras variables de entorno



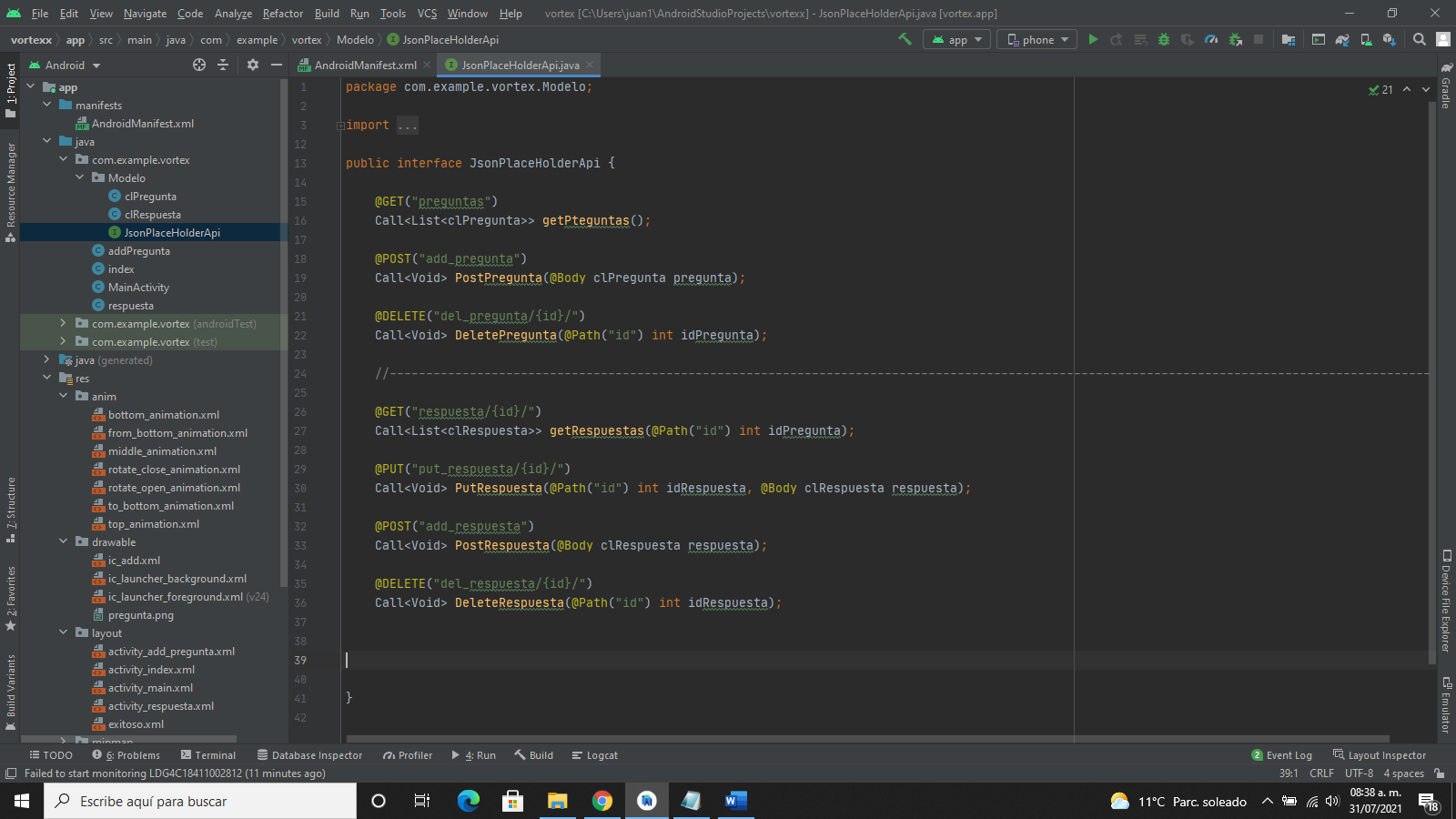
1. Creamos el archivo config.js para configurar la conexión y los puertos del servidor



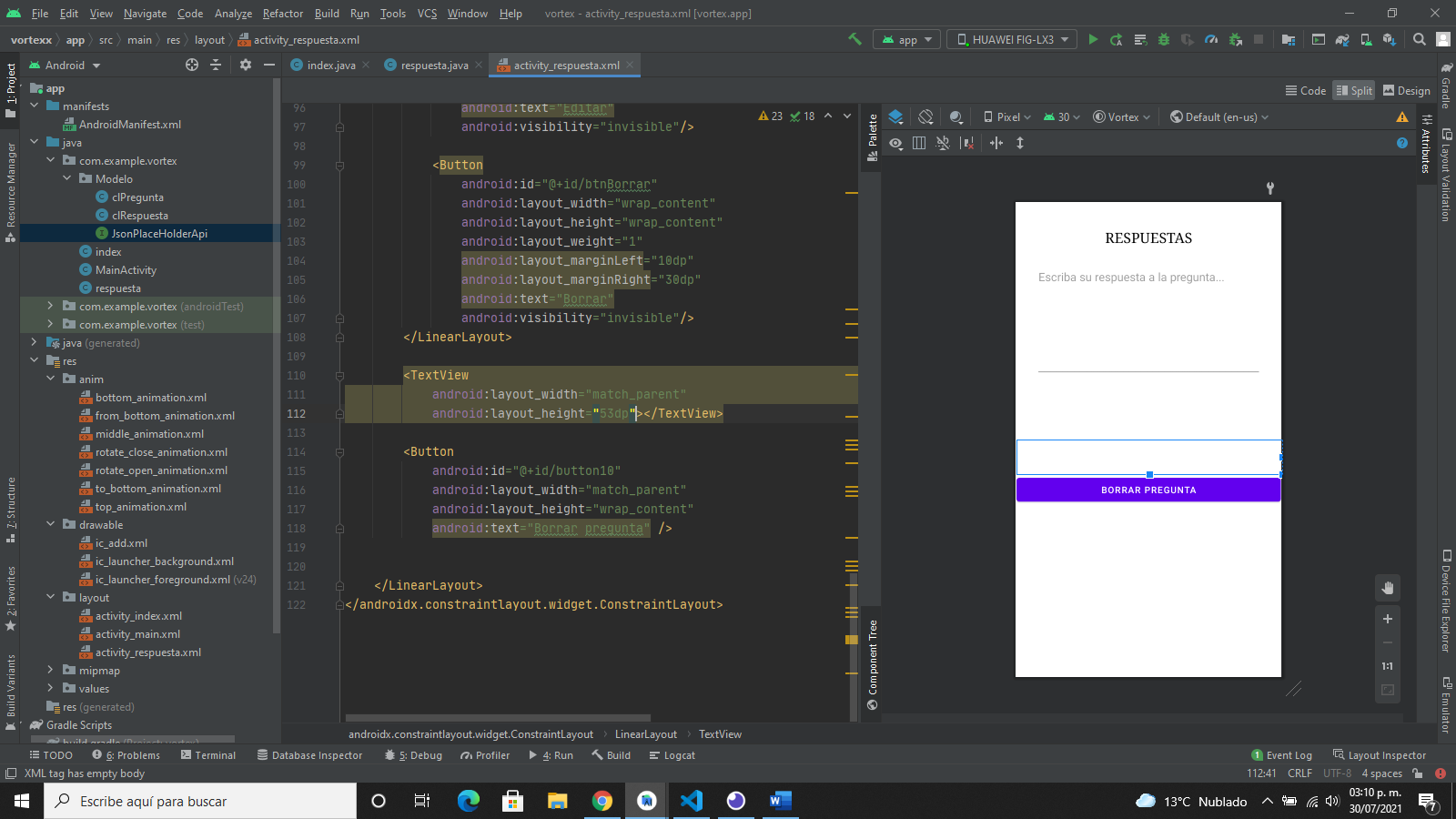
1. Creamos el archivo index.js el cual será el punto de entrada o ejecución del servidor



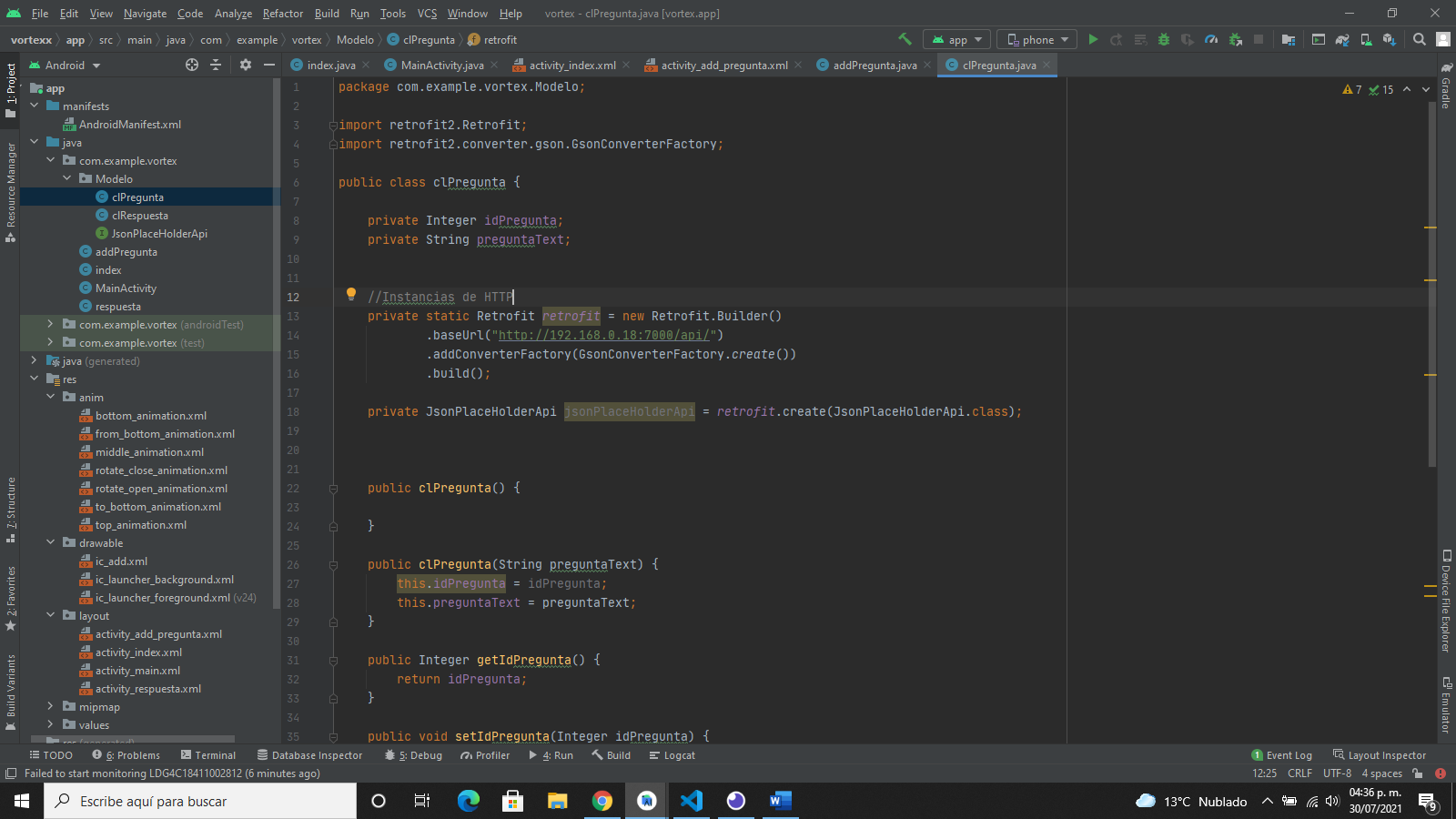
1. Creamos la app en Android Studio e instalamos la librería RETROFIT y CONVERTER-GSONpara realizar las peticiones HTTP CLIENT, creamos un archivo interfaz y en este hacemos las peticiones (GET, POST, PUT, DELETE)

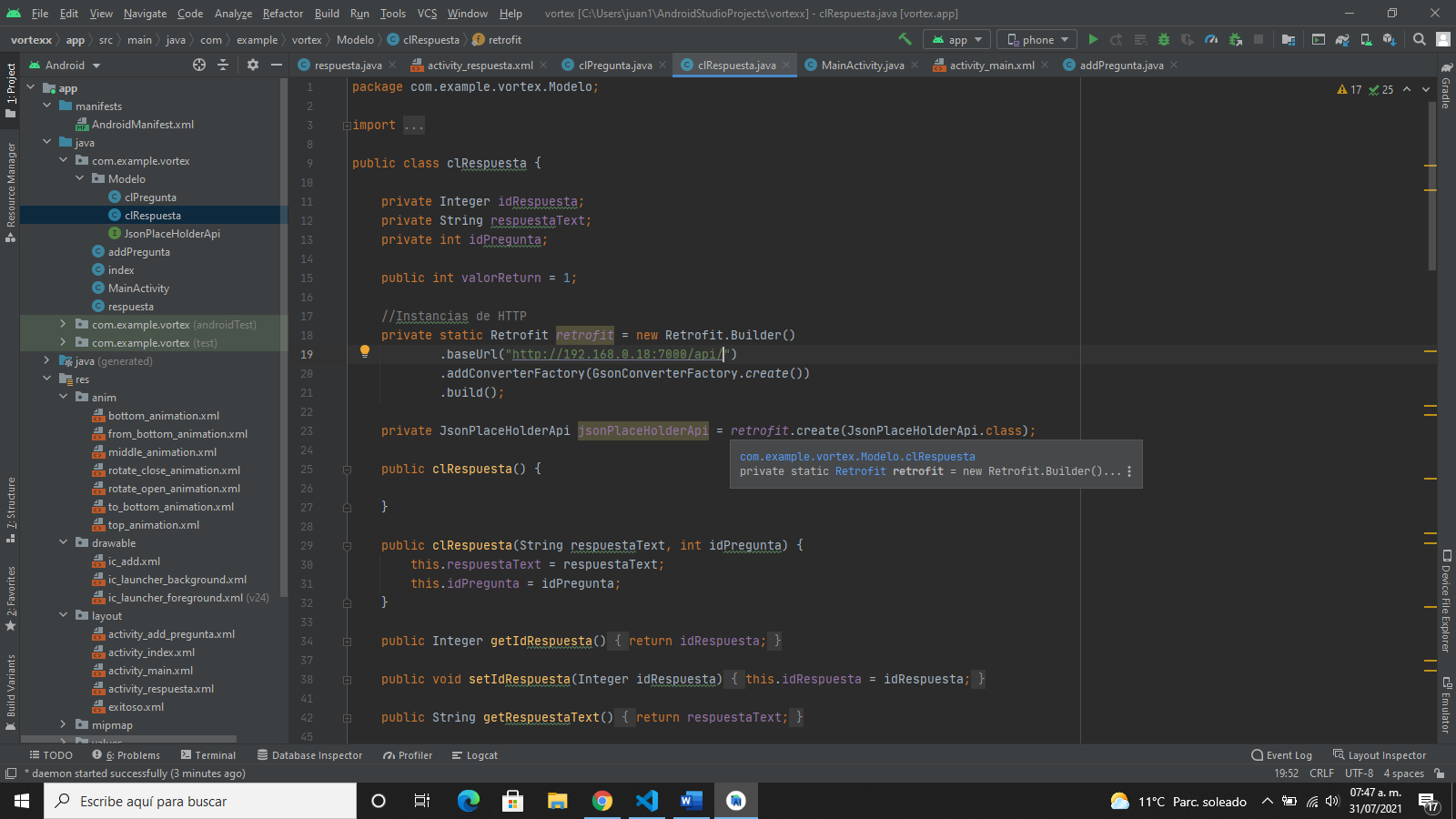


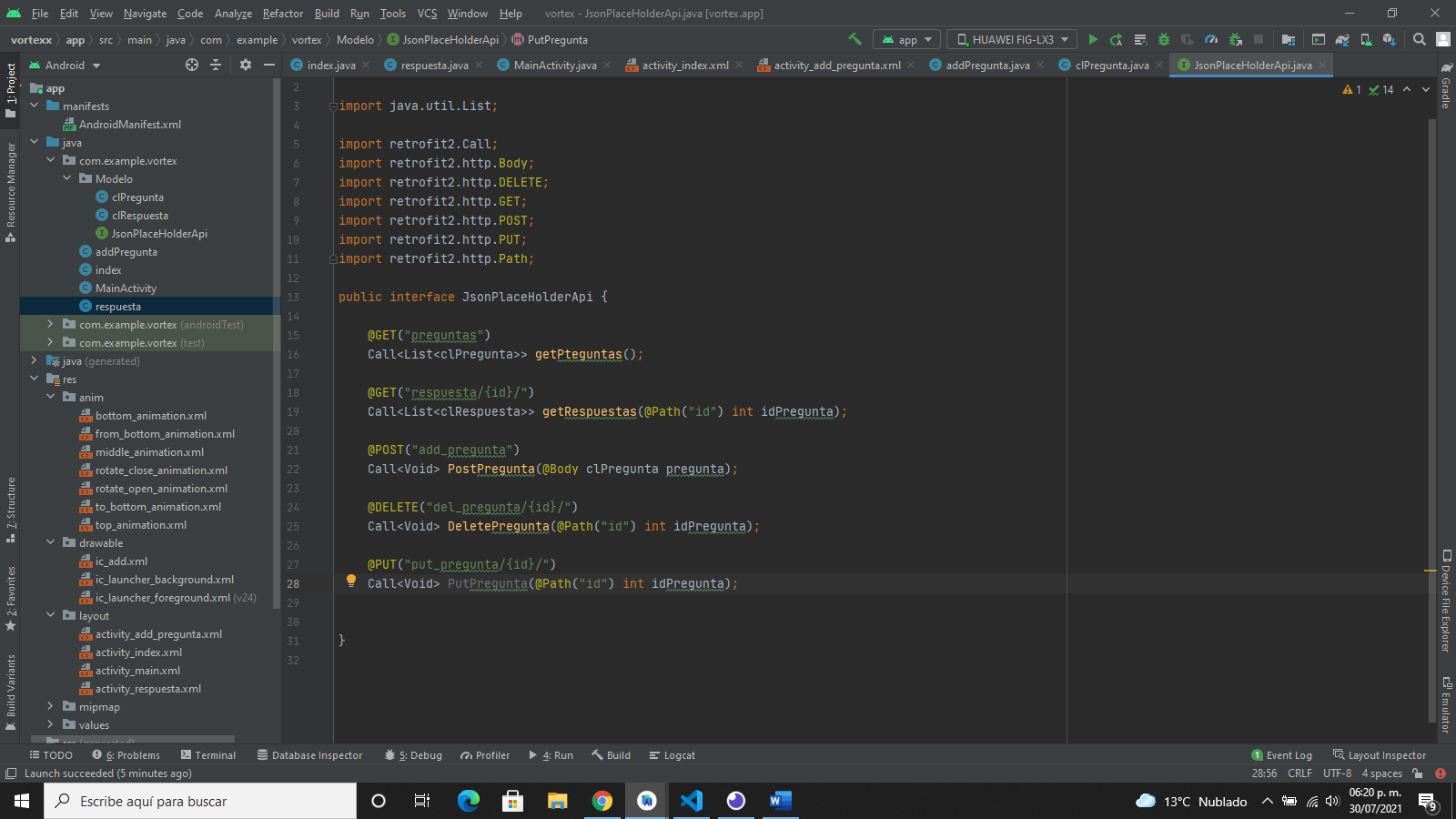
1. Creamos los formularios en Android

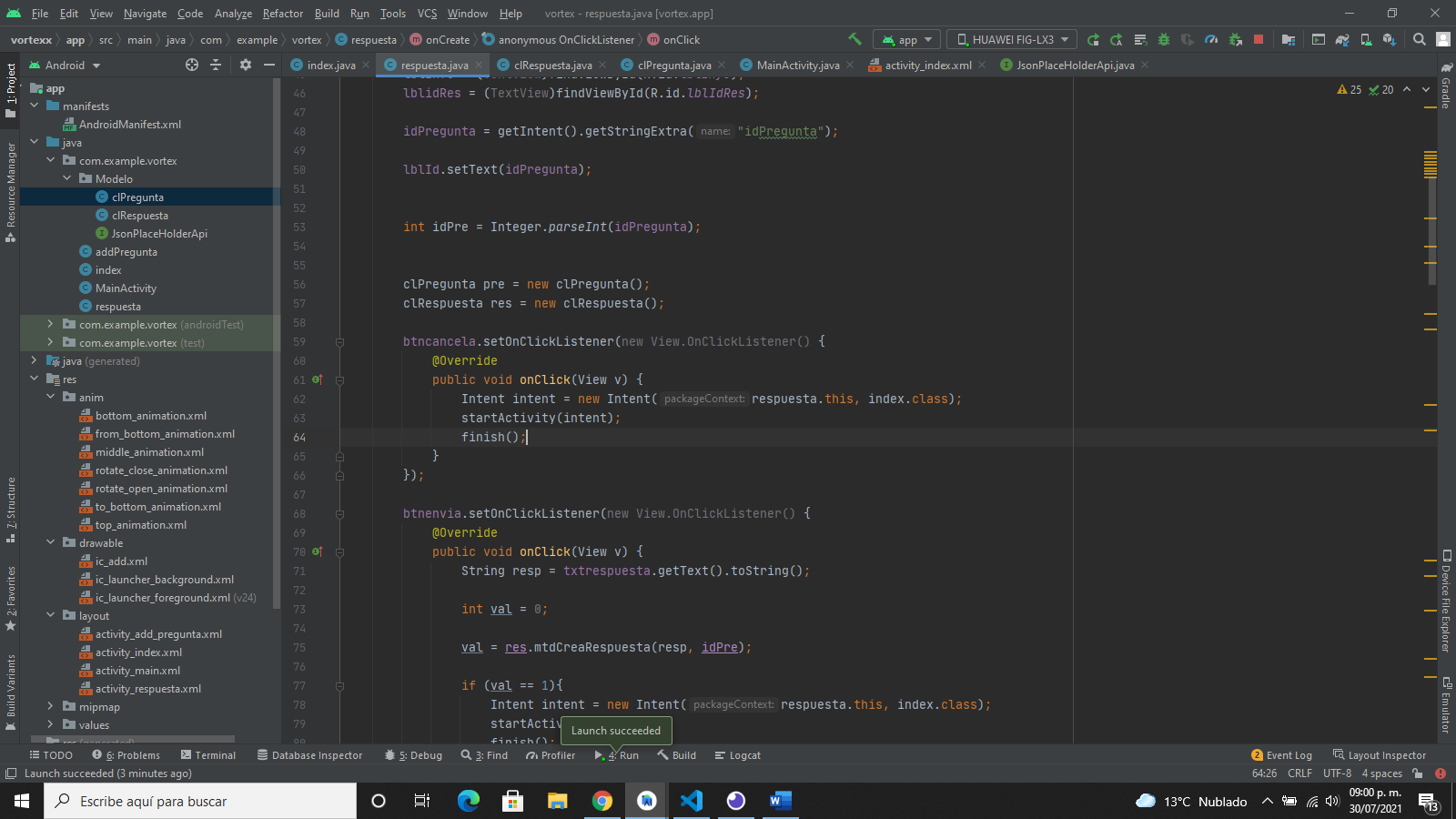


1. Creamos instancias de retrofit le asignamos la url para su consumo





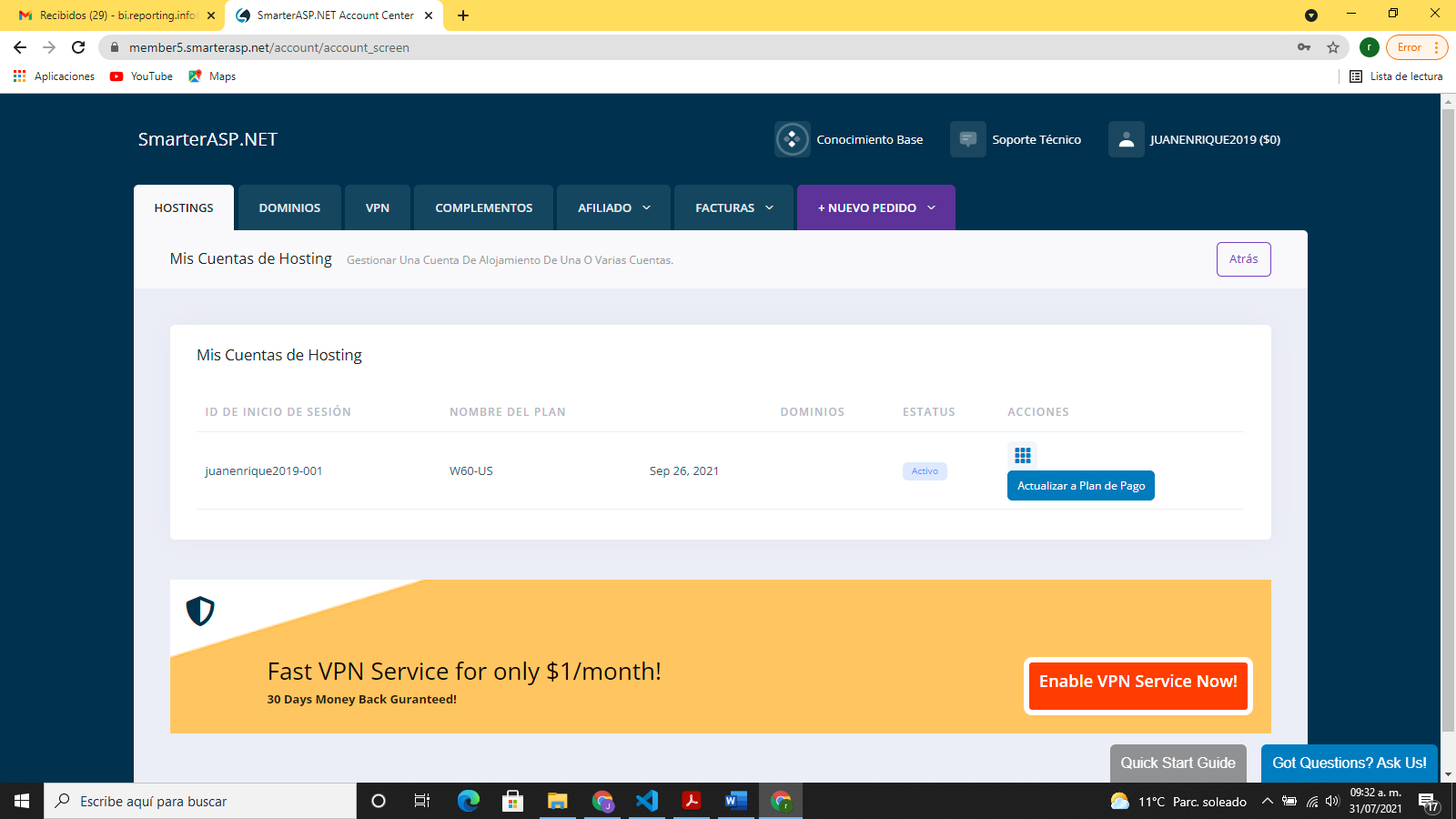


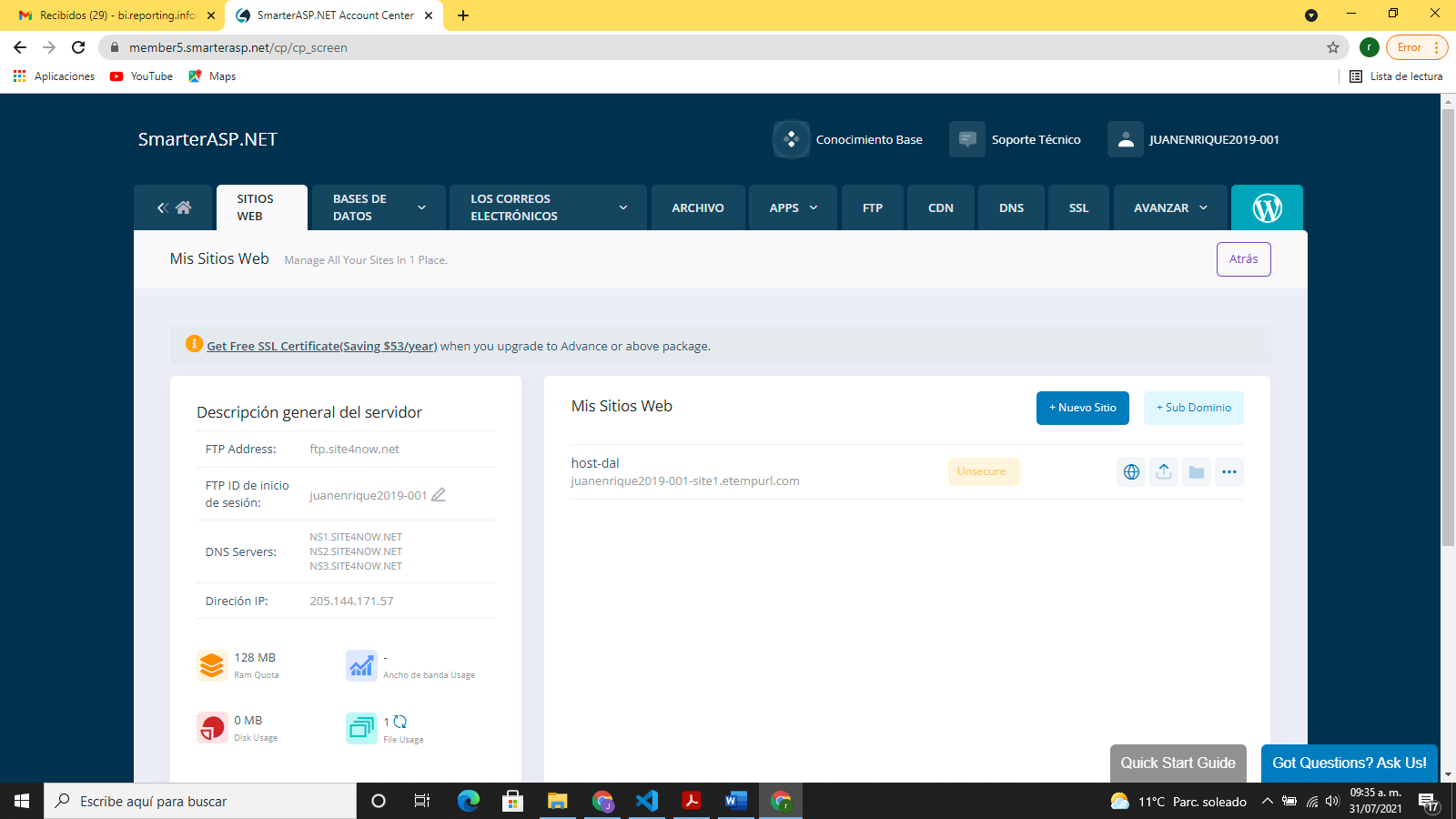


1. Desplegamos la BD en un Host gratuito llamado smarterasp.net

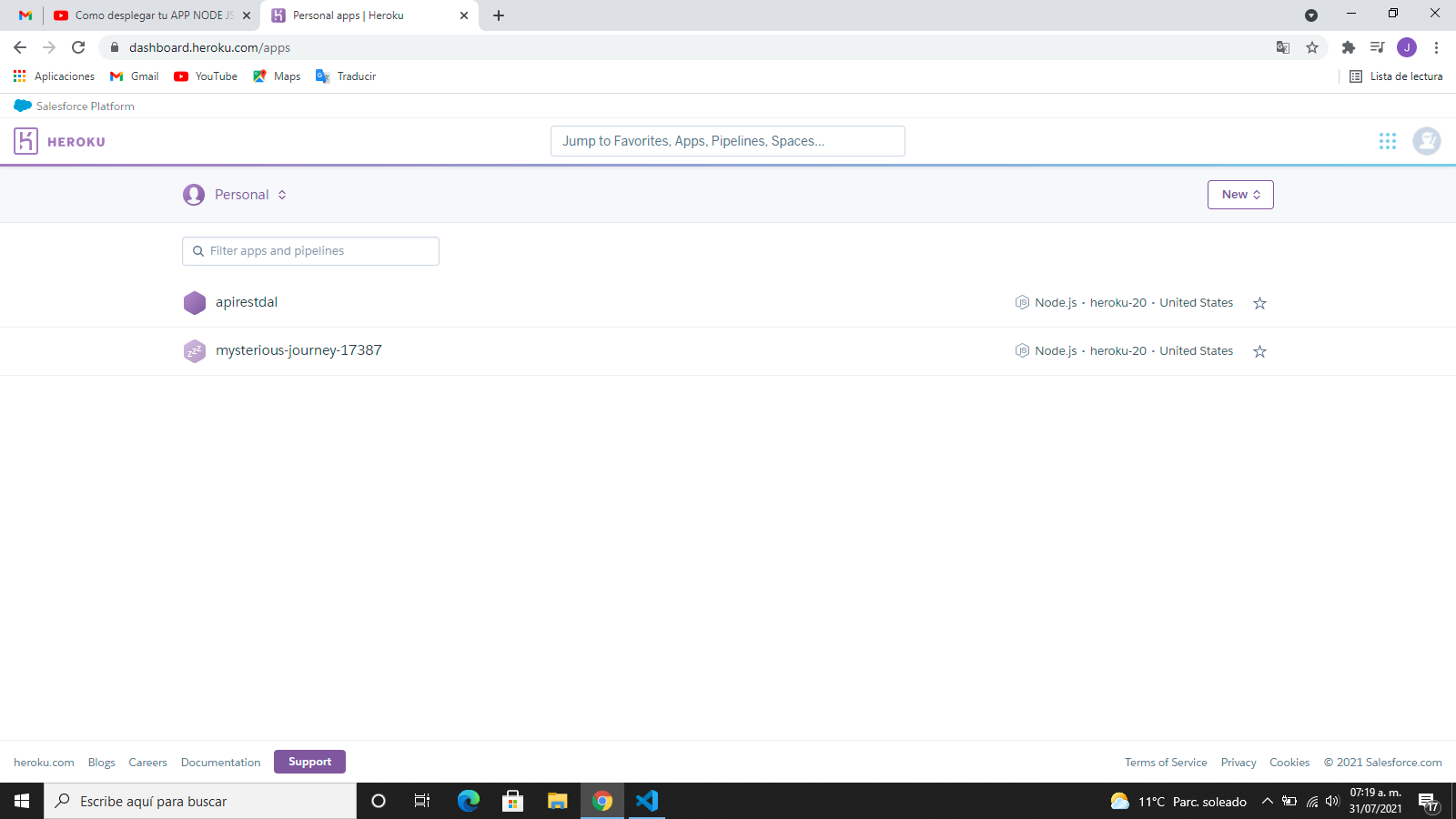
Su url:

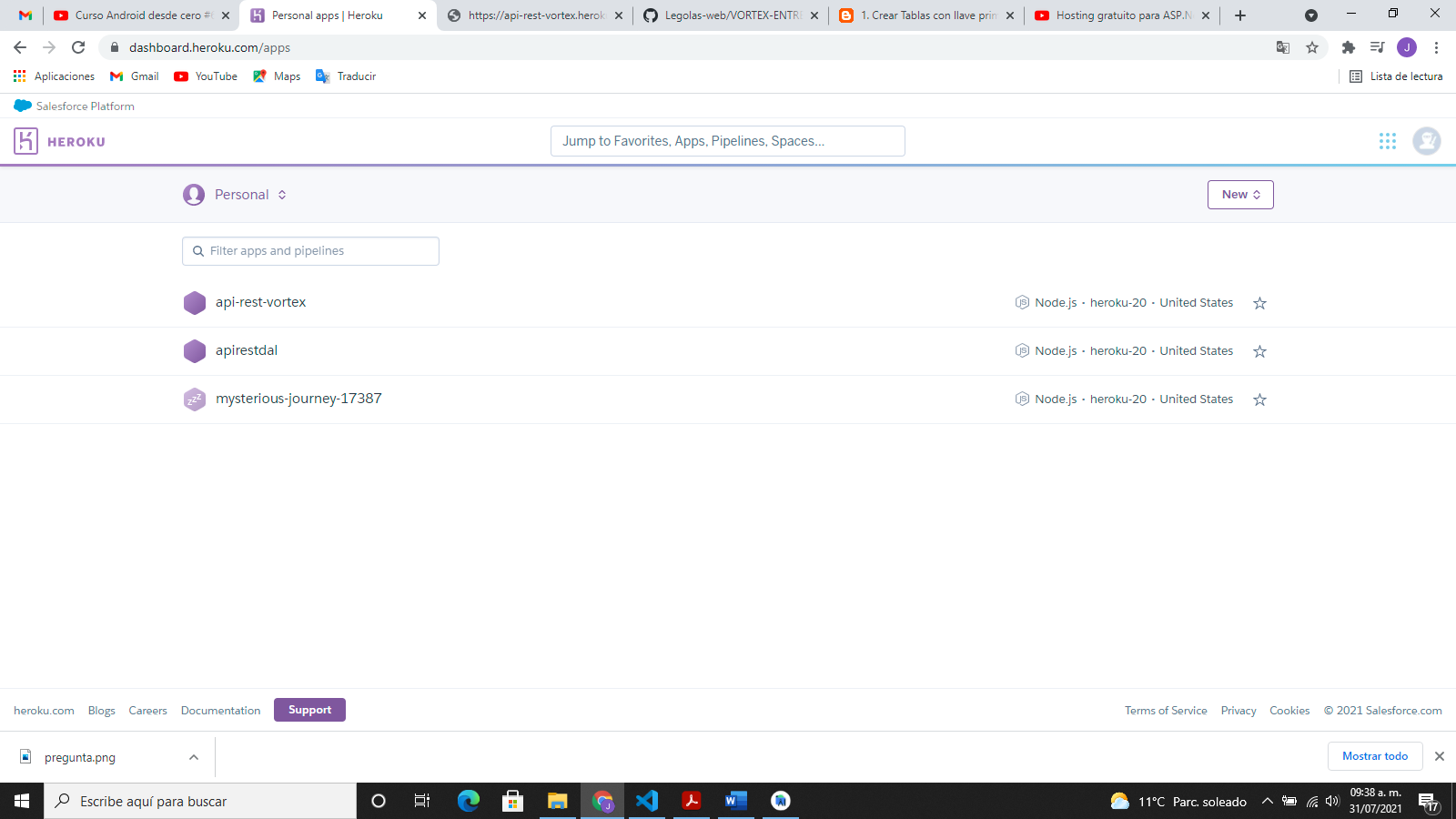
<https://member5.smarterasp.net/account/account_screen>

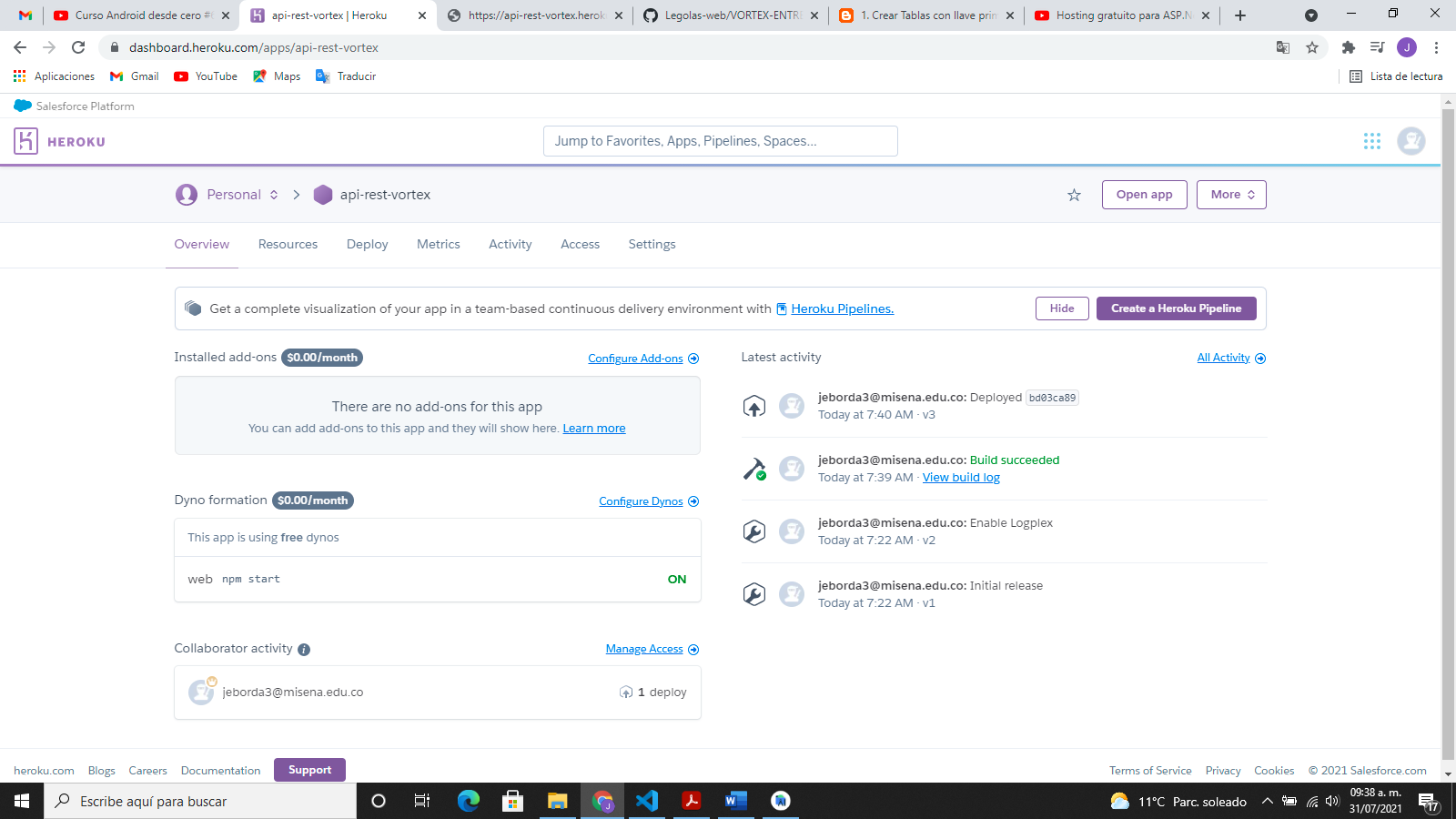


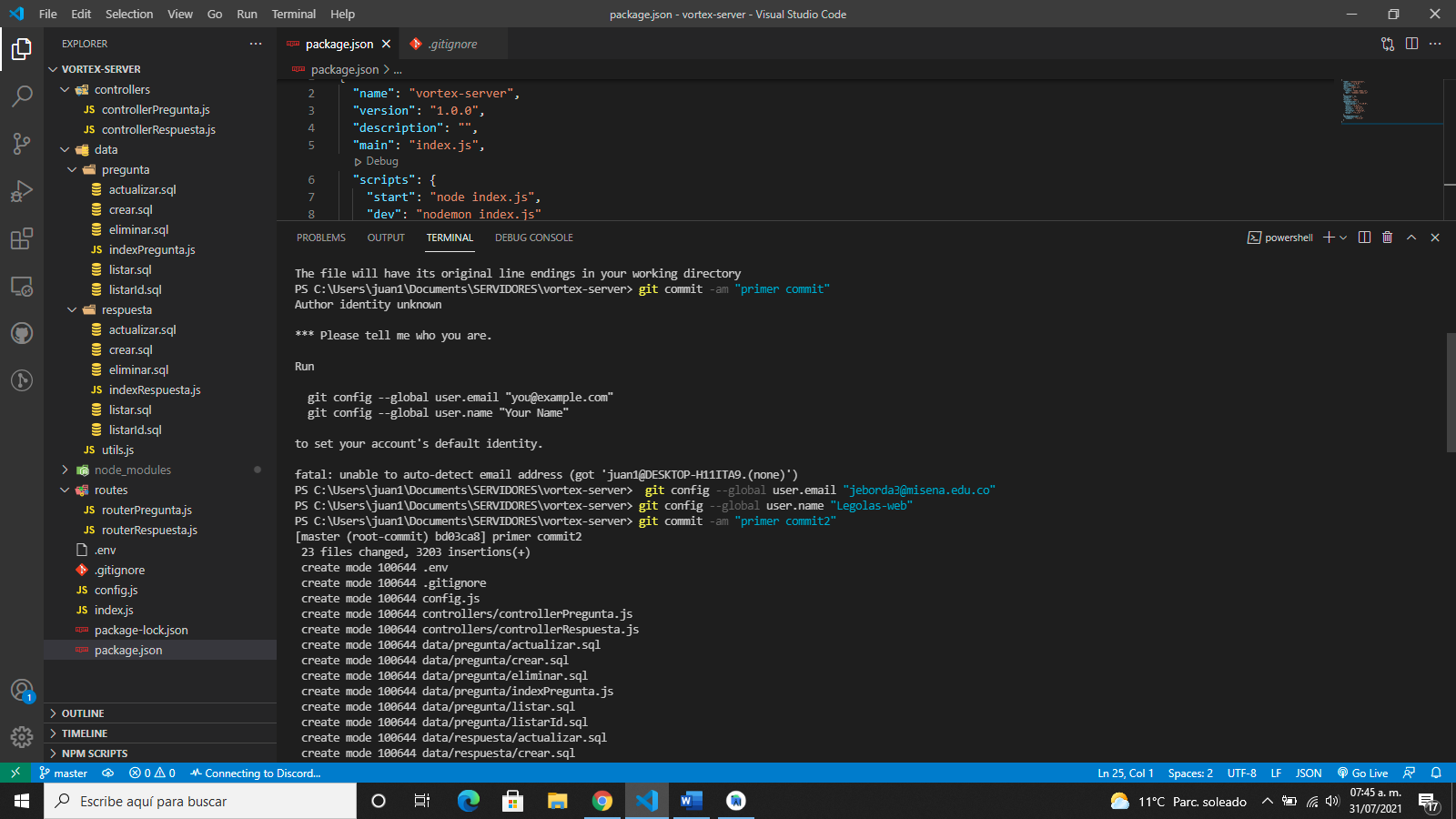


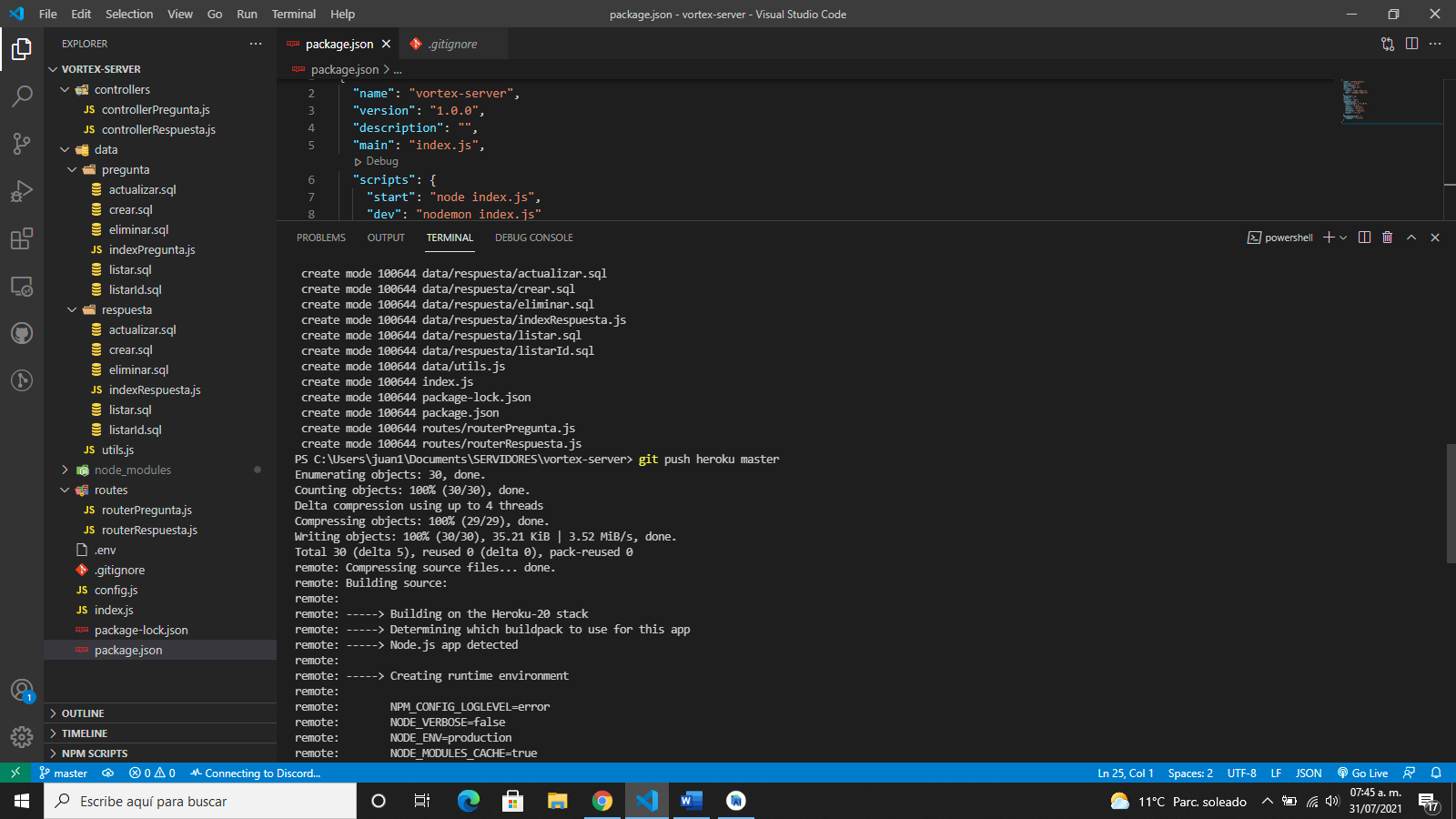
1. Desplegamos el BACKEND de node.js en HEROKU el cual es gratuito

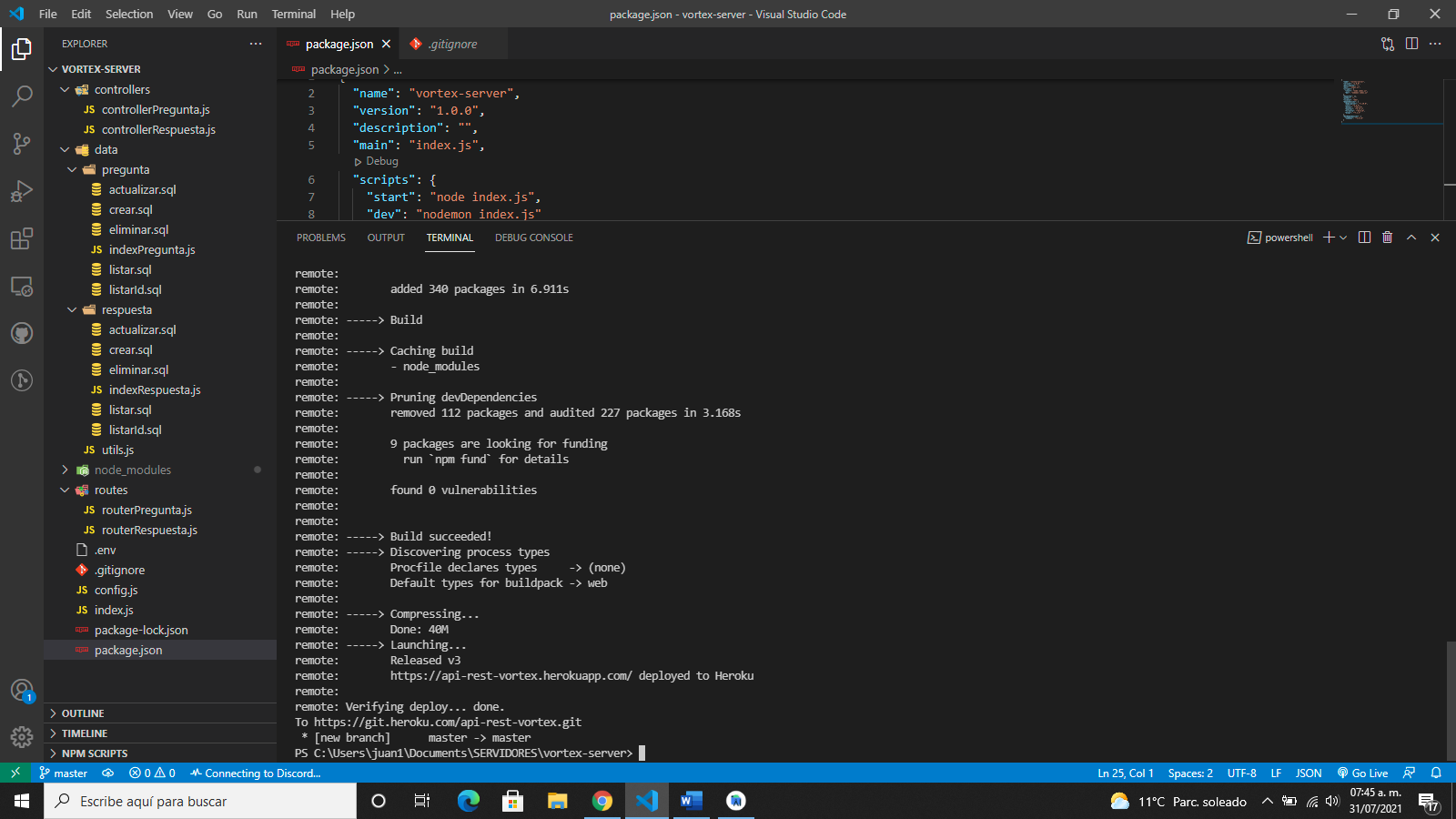






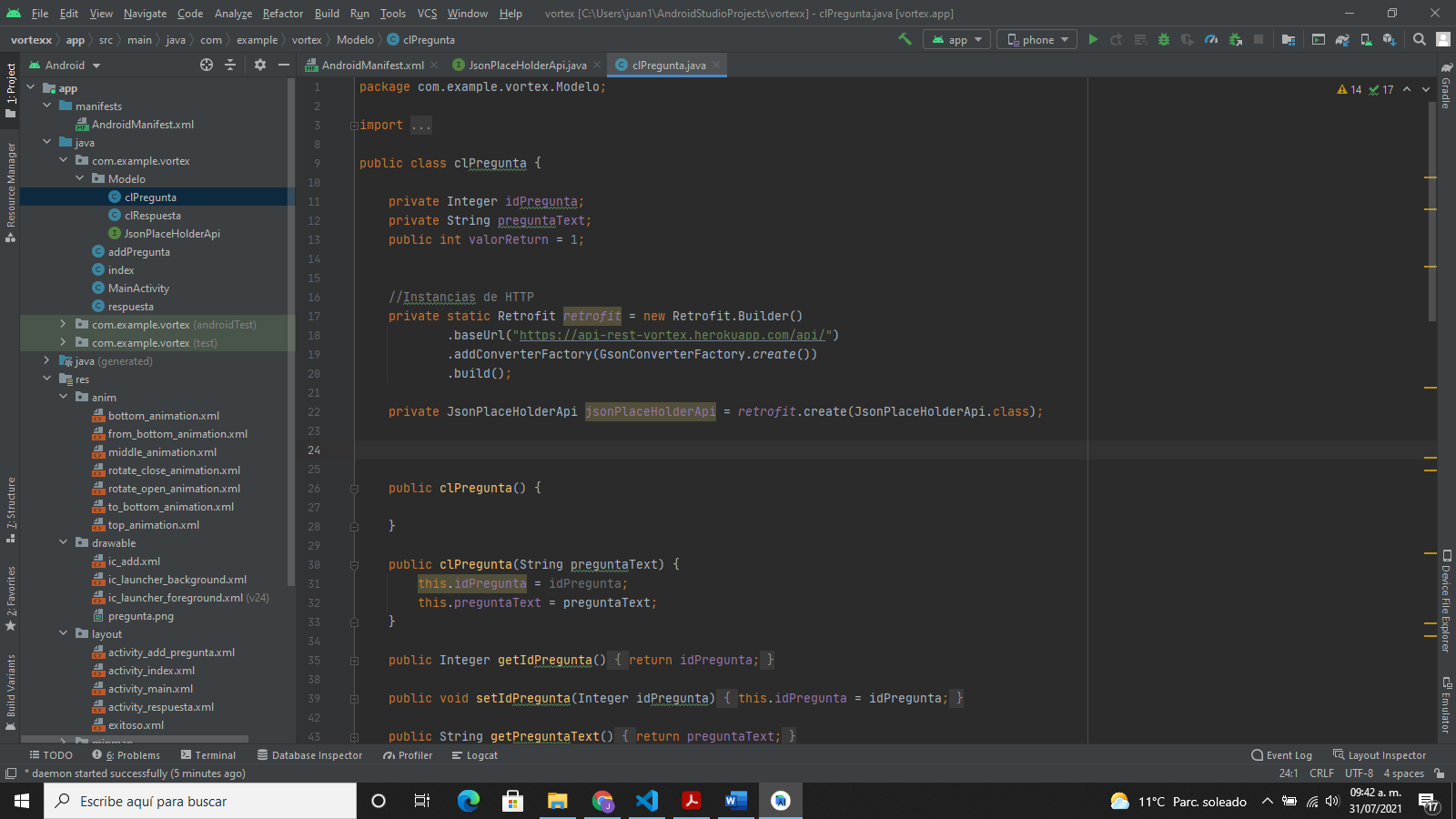


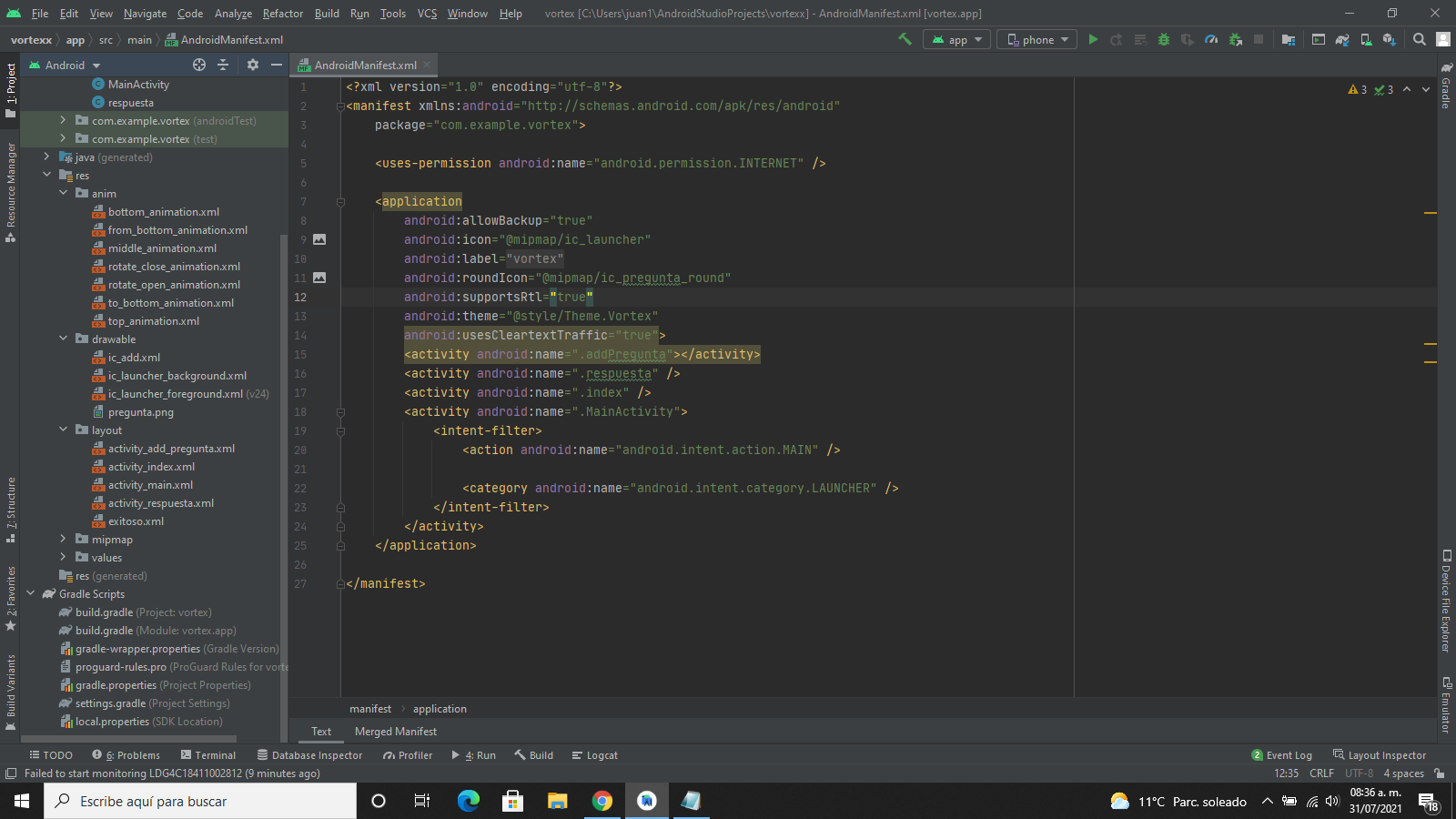




url api BACKEND: <https://api-rest-vortex.herokuapp.com/api/>

1. Cambiamos url a las instancias de RETROFIT y ponemos un icono a la app





1. Creamos el instalador de la app en Android este será un archivo .apk

Nombre del servidor sql 🡺 sql5105.site4now.net

Nombre del Usuario 🡺 db\_a77f55\_dal\_admin

Password user 🡺 00000111

URL API 🡺 <https://api-rest-vortex.herokuapp.com/api/>

ROUTES RESPUESTA

(GET) 🡺 <https://api-rest-vortex.herokuapp.com/api/respuestas>

(GET) 🡺 <https://api-rest-vortex.herokuapp.com/api/respuestas/id>

(POST) 🡺 [https://api-rest-vortex.herokuapp.com/api/add\_respuesta](https://api-rest-vortex.herokuapp.com/api/add_respuestas)

(PUT) 🡺 <https://api-rest-vortex.herokuapp.com/api/put_respuesta/id>

(PUT) 🡺 <https://api-rest-vortex.herokuapp.com/api/del_respuesta/id>

ROUTES PREGUNTAS

(GET) 🡺 <https://api-rest-vortex.herokuapp.com/api/preguntas>

(GET) 🡺 <https://api-rest-vortex.herokuapp.com/api/preguntas/id>

(POST) 🡺 <https://api-rest-vortex.herokuapp.com/api/add_pregunta>

(PUT) 🡺 <https://api-rest-vortex.herokuapp.com/api/put_pregunta/id>

(PUT) 🡺 <https://api-rest-vortex.herokuapp.com/api/del_pregunta/id>