**ADMINISTRACIÓN DE SISTEMAS Y SERVICIOS EN RED**

**SEGUNDA EVALUACIÓN PRÁCTICA - I TÉRMINO 2021**

**PARALELO 101**

Informe de la Segunda Evaluación Práctica

Nombre de la profesora:

Msig. Adriana Collaguazo Jaramillo

Número de grupo: 4

Nombres/Apellidos:

* Daniel Salinas
* Nathaly Indacochea

**ESCENARIO**

Debido a la gran demanda de aplicaciones web en la ciudad de Guayaquil. La empresa ADITA S.A. fue contratada por la M.I. Municipalidad de Guayaquil para crear una aplicación que monitoree la propagación del COVID-19 en la ciudad. La empresa ADITA S.A. los ha contratado para la ejecución e implementación de varios servicios que se necesitan de urgencia. A continuación se detallan los servicios que se deben implementar:

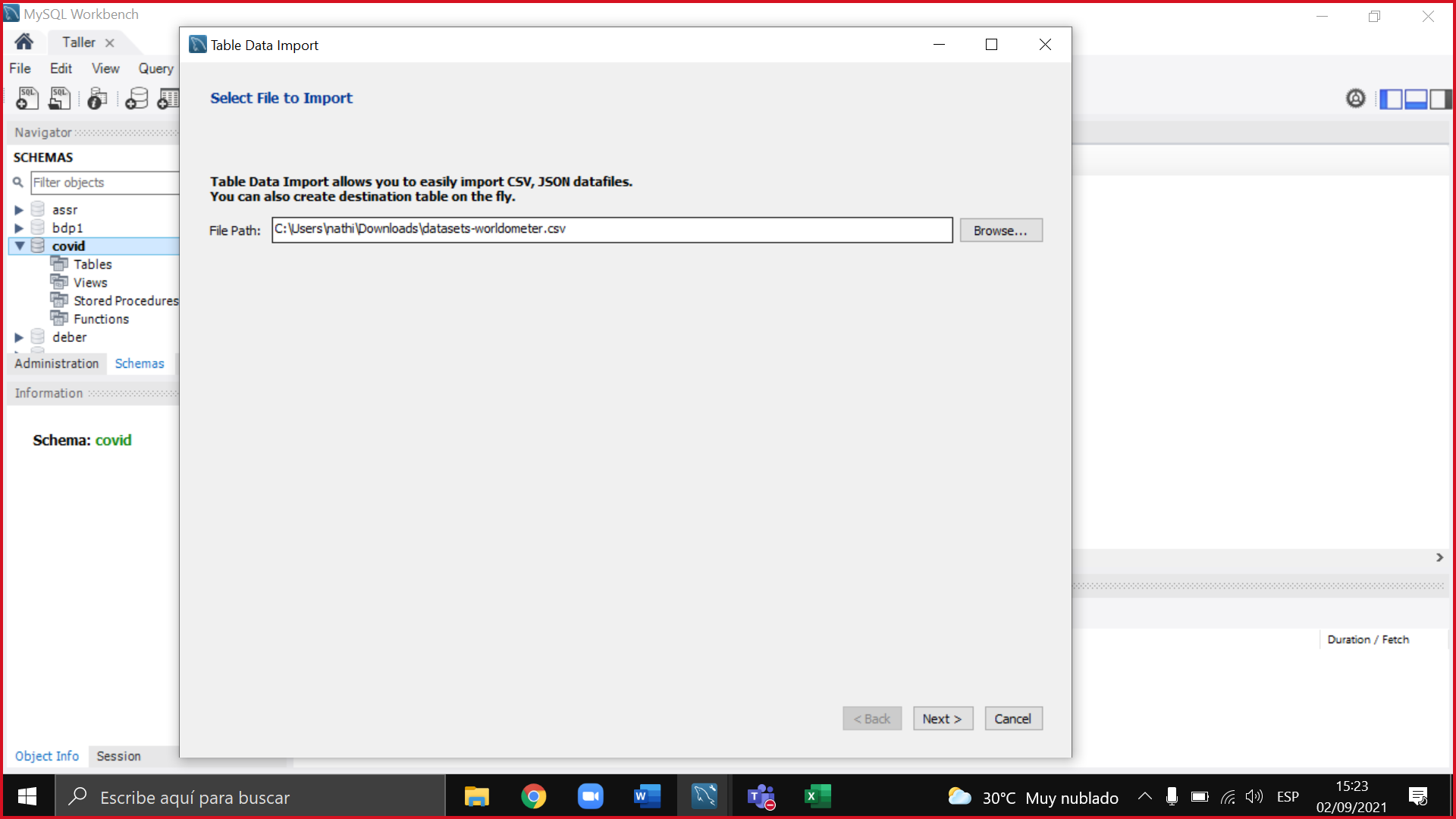
**TEMA #1: BASES DE DATOS (40 PUNTOS)**

* ADITA S.A. ha indicado que le gustaría trabajar con bases de datos relacionales.
* Crear una base de datos no relacional (noSQL) con el nombre “COVID”.
* Crear los objetos necesarios, tomando como referencia el conjunto de datos proporcionado en “datasets-worldometer.csv”.
* Mostrar la importación de todos los datos del archivo “datasets-worldometer.csv”.

**Nota: Colocar código fuente (en modo texto) e imágenes del proceso que evidencien el desarrollo del tema.**

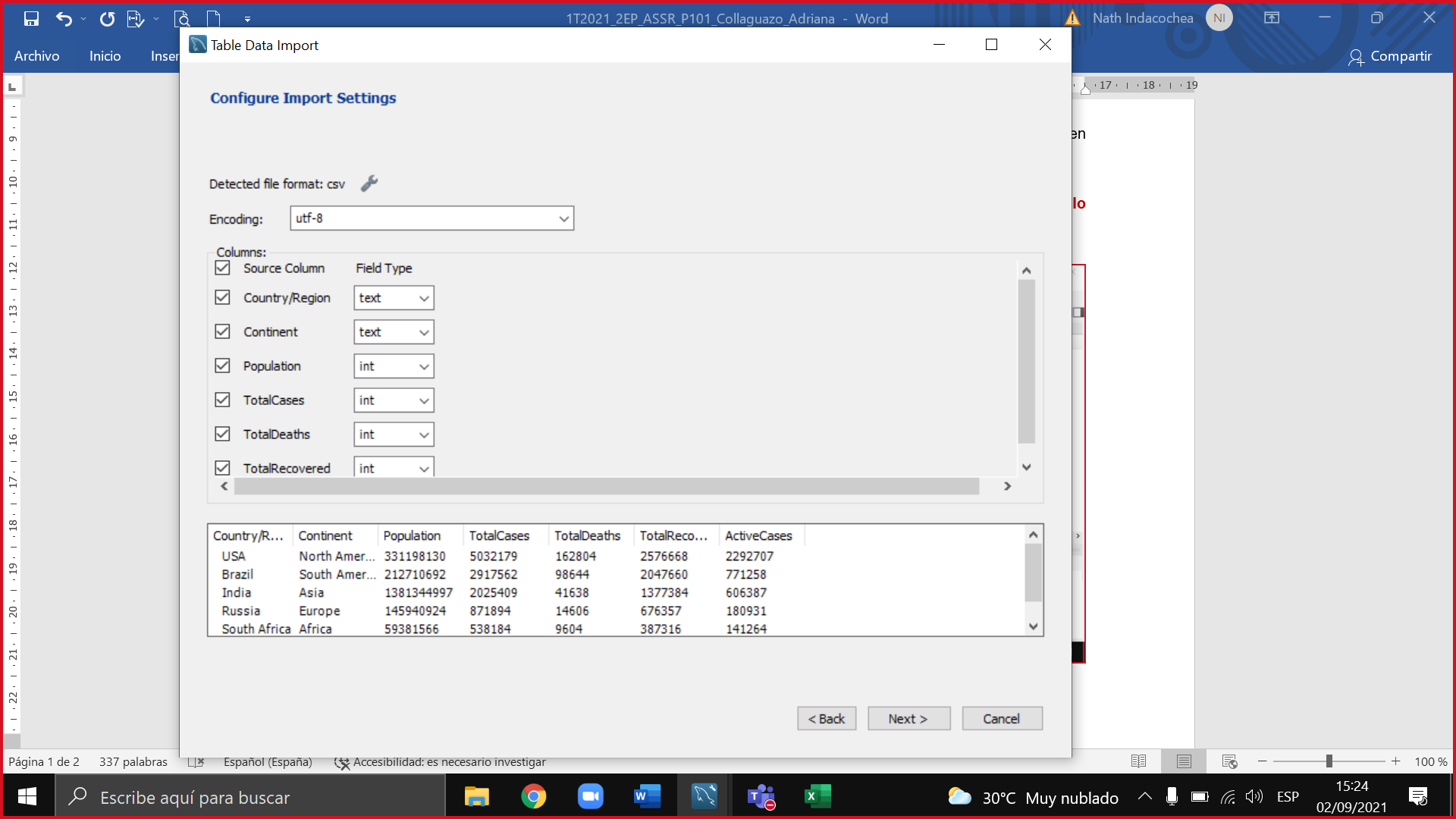
**PASO 1**

Importamos la base de datos no relacional llamada “datasets-worldometer.csv” con el comando en Workbench TABLE DATA IMPORT, seleccionamos la ruta y procedemos a su ejecución.



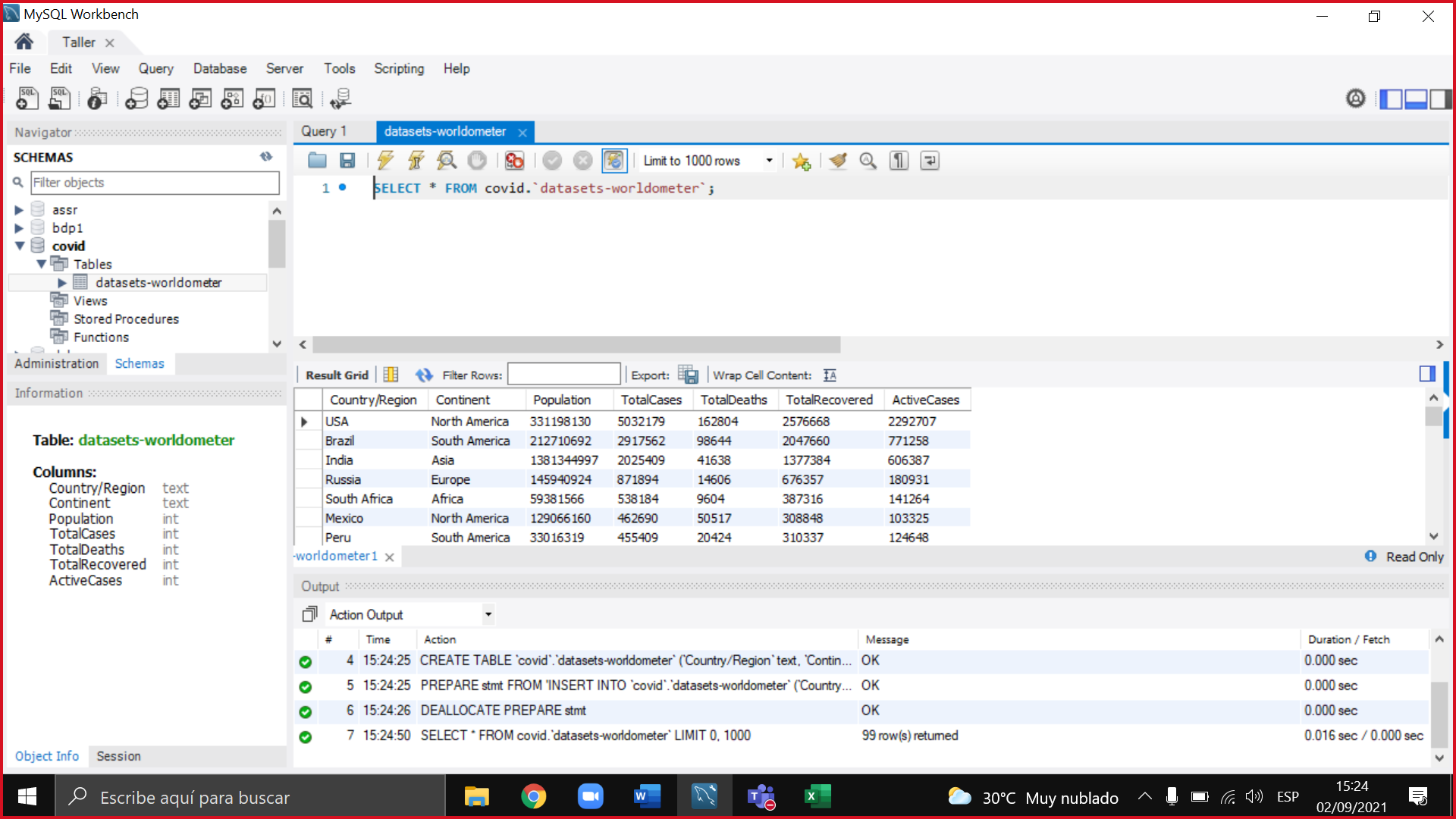
**PASO 2**

Configuramos los objetos necesarios para la creación de nuestra base de datos, en este caso seleccionamos todos, Country/Region, Continent, Population, TotalCases, Total Deaths, TotalRecovered, ActiveCases, y procedemos a importar nuestros datos con el tipo de dato que le corresponde.



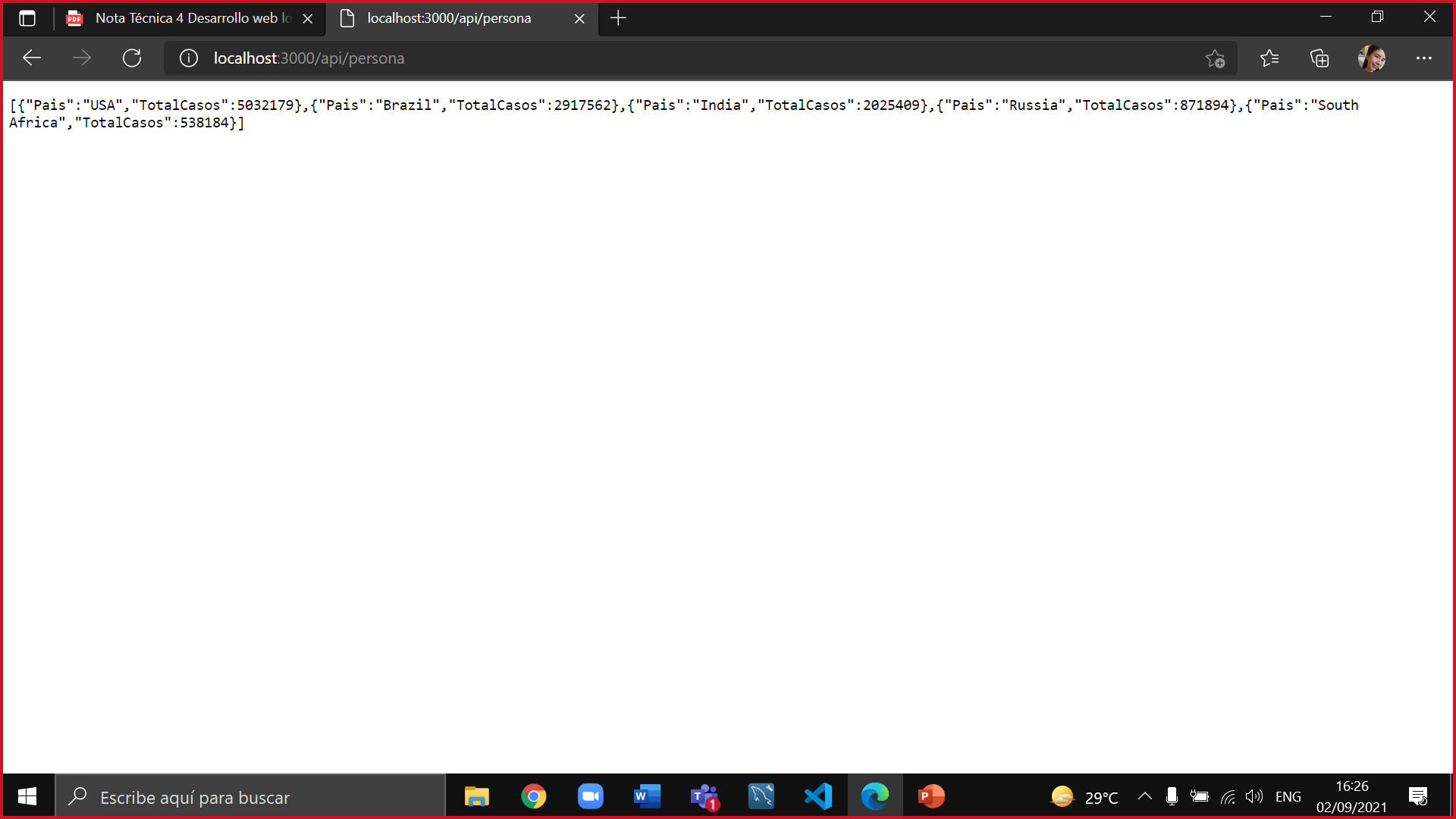
**PASO 3**

Verificamos la correcta importación de datos y la creación de nuestra tabla datasets-worldometer.



**TEMA #2: PÁGINA WEB (55 PUNTOS)**

* Presentar en una página web, el top de los 5 países que tienen el *menor número total de casos de COVID-19 (columna de TotalCases).*



**Nota: Colocar evidencia como el código fuente (en modo texto) e imágenes de la página web que evidencien la ejecución de la aplicación. Colocar el enlace del repositorio de GitHub de su código fuente. Además, presentar un video donde se muestre la aplicación web en funcionamiento.**

[***https://github.com/nindacoc/examenASSRP101/tree/master***](https://github.com/nindacoc/examenASSRP101/tree/master)

**databaseCon.js**

const mysql = require('mysql');

module.exports = () => {

 return mysql.createConnection({

 host: "localhost",

 user:'root',

 password: '12345678',

 database:'covid',

 });

 }

Se realiza la conexión especificando la base de datos (mysql.createConnection()), luego se agrega el host, el usuario, la contraseña, y el nombre de la base de datos creada.

Y se lo exporta como módulo, para luego poder usarlo en cualquier parte de nuestro programa, en este caso en el archivo persona.controller.js.

**Persona.controller.js**

const dbConnection = require('../config/databaseCon'); const connection = dbConnection();

let getPersona = async (req,res)=>{

await connection.query("select `Country/Region` as Pais, TotalCases as TotalCasos from `datasets-worldometer` Order by TotalCasos DESC Limit 5 ", (err,result)=>{ if (result)

res.send(result);

else

res.status(500).send(err);

});

}

let addPersona = async (req,res)=>{

const {Country,Continent,Population,TotalCases,TotalDeaths,TotalRecovered, ActiveCases  } = req.body

await connection.query(`INSERT INTO datasets-worldometer VALUES(${Country}, '${Continent}', '${Population}',, '${TotalCases}',, '${TotalDeaths}',, '${TotalRecovered}',, '${ActiveCases}')`, (err,result)=>{

if (result)

res.send({Country,Continent,Population,TotalCases,TotalDeaths,TotalRecovered, ActiveCases });

else

res.status(500).send(err);

});

}

module.exports = { getPersona, addPersona}

Se crean la función getPersona() que nos dará los registros de nuestra base de datos que ya ha sido exportada, y se va a validar una respuesta. También se crea la función addPersona(), que puede ser usada por en cualquier parte del código añadiendo una persona con sus valores específicos a nuestra tabla.

**persona.js**

const {Router} = require("express"); const router = Router();

const{ getPersona, addPersona } = require("../controller/persona.controller");

router.get('/', getPersona); router.post("/",addPersona);

module.exports = router;

**app.js**

const express = require('express'); const app = express();

// body-parser app.use(express.json());

app.use(express.urlencoded({extended:false})); app.set('port', process.env.PORT | 3000);

app.get('/', (req,res)=>{

res.send("Bienvenido a mi API creado con la profesora Adriana Collaguazo");

});

app.use("/api/persona",require("./routes/persona") );

app.listen(app.get('port')); console.log(`Server on port ${app.get('port')}`)