UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
SISTEMAS DE BASES DE DATOS 1
ING. LUIS ESPINO
AUX. JONNATHAN CASTILLO

Manual Tecnico

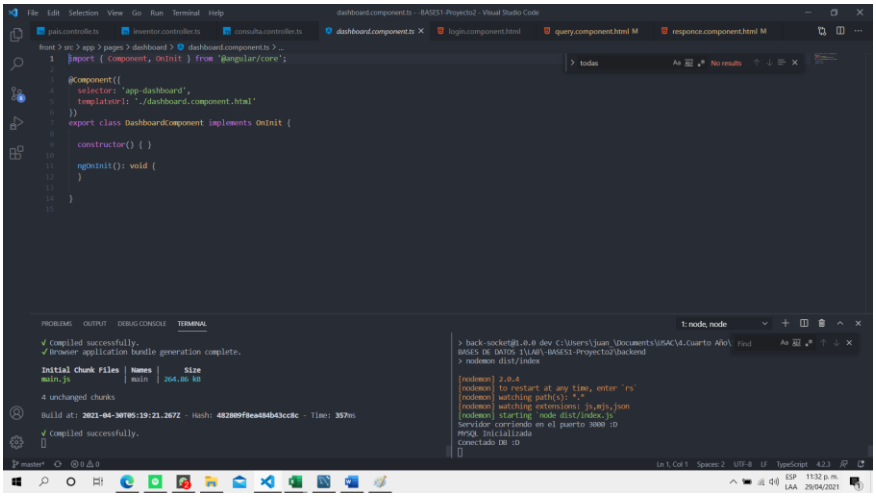
JUAN JOSE RAMOS CAMPOS
201801262
03/05/21

Contents

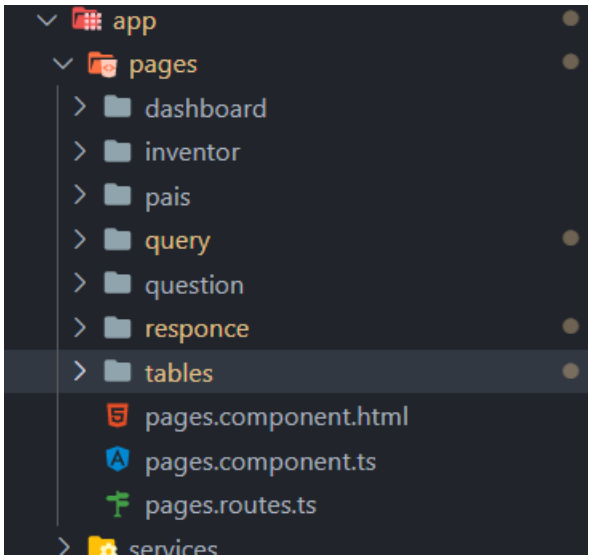
FrontEnd

Para el desarrollo del frontend del proyecto se utilizo Angular versión 7,

para las diferentes vistas se diseñaron componentes varios que permite la funcionaldad de la misma.



En la siguiente imagen se muestra el componente del dashboard dashboard, de igual manera se diseñaron los demás componentes, separando la aplicación en módulos que da un orde u estructura para la misma



Rutas

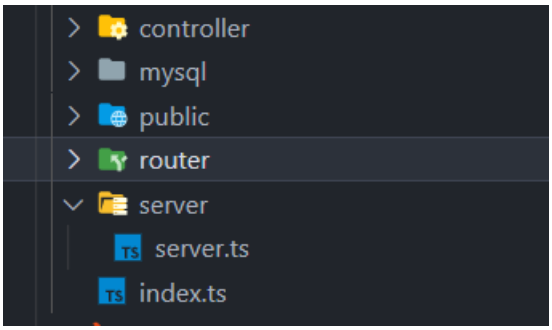
Angular funciona a base de rutas, para poder moverse dentro de la aplicación se dividio en rutas a las que se accede desde cada componente, dichas rutas con las siguientes

```
export const page_routes: Routes = [
    //{ path: 'admin', component: AdminnComponent },
    { path: 'home', component: DashboardComponent },
    { path: 'tables', component: TablesComponent },
    { path: 'query', component: QueryComponent },
    { path: 'country', component: PaisComponent },
    { path: 'question', component: QuestionComponent },
    { path: 'responce', component: ResponceComponent },
    { path: 'inventos', component: InventorComponent },

    {path: '**', pathMatch: 'full', redirectTo: 'admin'}
  ];
```

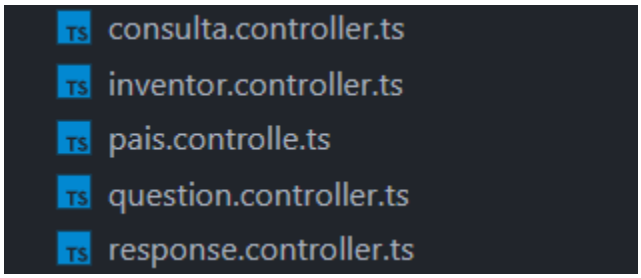
Backend

Para el desarrollo del backend se utilizo Nodejs + Express

El servidor se dividio también en módulos siendo los siguientes



Controller



Cuenta con varios controladores que permiten la ejecución de las acciones hechas en desde el fronend, por ejemplo, el controlador de consulta recibe el id del a consulta y dependiendo del mismo selecciona una u otra y se envia a ejecutar

```
public static getInstance() {
    return this._instance || ( this._instance = new this() );
}

query = (req: Request, res: Response) => {
    var id = req.params.id;
    var query = "";

    if (id == "1") {
        query =
        "SELECT profesional, P.nombre AS NOMBRE_PROFESIONAL, COUNT(*) as inventos_asignados FROM Asignacion_Inv
        "JOIN Profesional P on P.id = AI.profesional " +
        "GROUP BY profesional " +
        "ORDER BY inventos_asignados desc; ";

    }
    else if (id == "2") {···
    }
    else if (id == "3") {···
```

Por su parte, el controlador de país, por ejemplo, tiene las funciones necesarias para el CRUD

```
postPais = (req: Request, res: Response) => {

    console.log(req.body)

    const { country, poblation, area, capital, region } = req.body;

    var query = "INSERT INTO Pais(nombre, poblacion, area, capital, region) "+
    " VALUES(?, ?, ?, ?, ?);";
    MySQL.sendQuery(query, [country, poblation, area, capital, region], (err:any, data:Object[]) => {
        if(err) {
            res.status(400).json({
                ok: false,
                status: 400,
                error: err
            });
        } else {

            res.json(data)
        }
```

En la foto anterior se mostro el método para agregar nueva info de países a la tabla

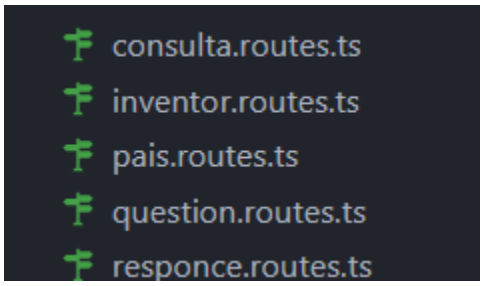
Mysql

Este modulo guarda la clase conexión, que permite como dice su nombre, crear la conexión con la base de datos

```
export default class MySQL {
    private static _instance: MySQL;

    connection: mysql.Connection;
    state: boolean = false;

    constructor() {
        console.log("MYSQL Inicializada");
        this.connection = mysql.createConnection({
            host: 'localhost',
            user: 'root',
            password: '1234',
            database: 'Proyecto2'
        });

        this.conectarDB();
    }

    public static getInstance() {
        return this._instance || ( this._instance = new this() );
```

Guarda un objeto con la cadena de conexión, el host, el usuario, el password y la base de datos a la que se va a conectar.

Router



Guarda las clases que contienen las diferentes rutas de la aplicación, por ejemplo las rutas de país se muestran a continuación

```
import { Router } from "express";
import controller from "./../controller/pais.controlle"
const router = Router();

router.get('/pais/get_all', controller.getInstance().getAll);
router.get('/pais/get/:id', controller.getInstance().getPais);
router.get('/pais/region', controller.getInstance().getRegion);
router.post('/pais/add', controller.getInstance().postPais);
router.delete('/pais/delete/:id', controller.getInstance().deletePais);
router.put('/pais/update/:id', controller.getInstance().updatePais);
```

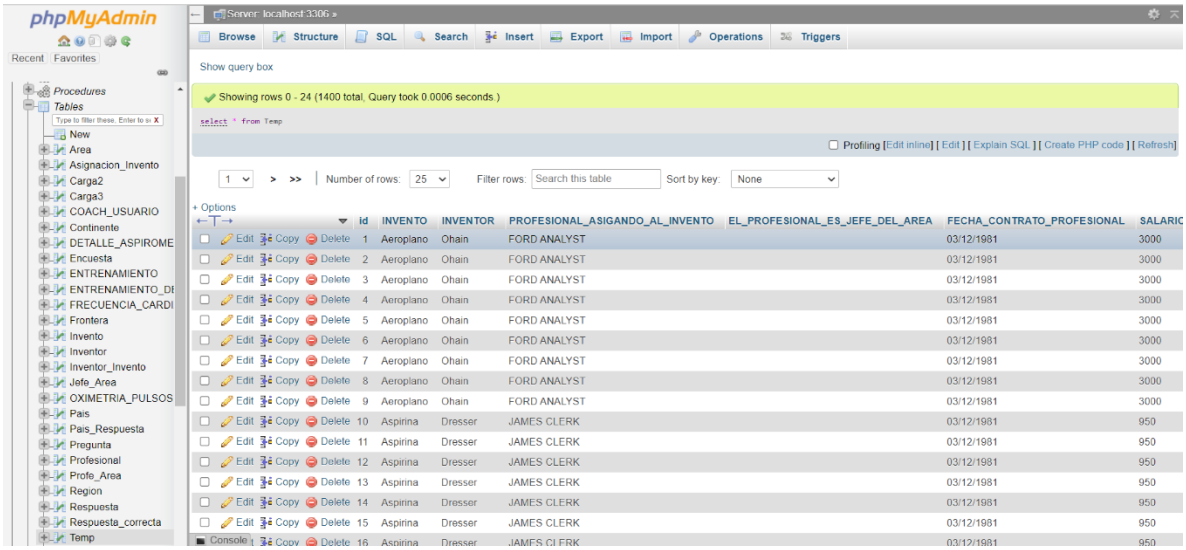
Dependiendo de la ruta que llegue, así se accede a cada uno de las funciones del controlador de país.

Index

Es la clase principal, dado que es el servidor como tal, aquí se declaran los diferentes elementos como el puerto, los headers, las rutas, etc.

MySQL

Para almacenar la información se utilizo un servidor de mysql, se accede mediante phpmyadmin



En el servidor se hizo la carga de archivos y mediante los scripts correspondientes se hizo la distribución de la información a las diferentes tablas.

Heroku

Para subir el servidor del proyecto se utilizo heroku, para ello se genero el build de la aplicación y con el comando “npm run build” y se subió la carpeta dist