Manual Técnico

Se necesita tener instalado

Node Js

Angular

Server

Correo: nodemailer

Imágenes: multer

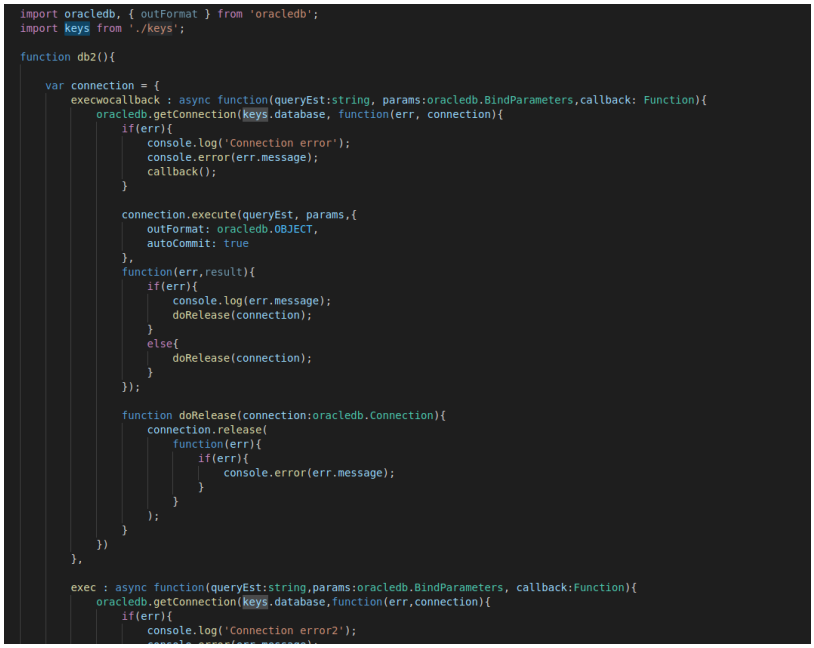
Incriptacion de contraseña: crypo (funcion md5)

La parte de la Api se programa en el Index.js

```
public app: Application;
public server : Server;
public io : SocketIO.Server;
constructor(){
    this.app = express();

    this.config();
    this.server = createServer(this.app);
    this.io = socketIo.default(this.server);
    this.routes();



}

config(): void{
    this.app.set('port',3009);
    this.app.use(morgan('dev'));
    this.app.use(cors());
    this.app.use(express.json());
    this.app.use(express.urlencoded({extended:false}));

}

routes():void {
    this.app.use('/', indexRoutes);
    this.app.use('/api', apiRputes);
    this.app.use('/user', userRoutes);
    this.app.use('/image', imageRoutes);
    this.app.use('/uploads', express.static(path.resolve('uploads')));
    this.app.use('/producto', productoRoutes);
}

start():void{
    var mensaje:string="asd";

    this.io.on('connection', (socket:Server) => {
        console.log('mas de algo envia');
  });

    this.server.listen(this.app.get('port'),() => {
        console.log('Server on port ',this.app.get('port'));
    });
```

OUTPUT DEBUG CONSOLE TERMINAL

Como se maneja la conexión de la base de datos

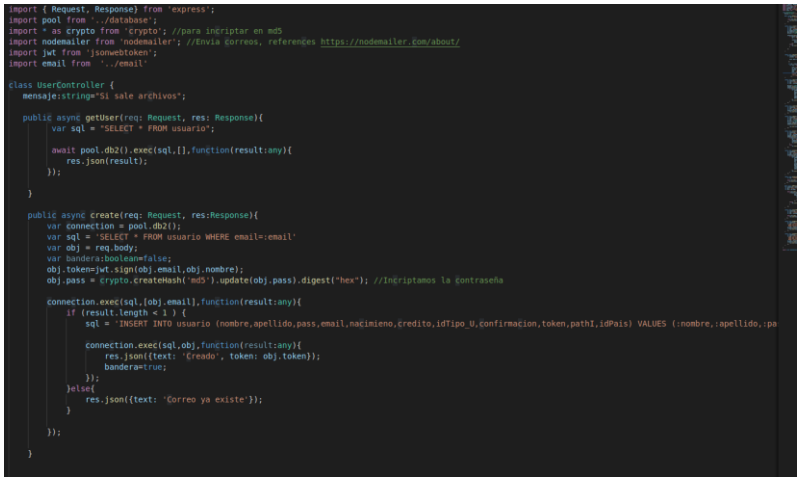


```
import oracledb, { outFormat } from 'oracledb';
import keys from './keys';

function db2(){

    var connection = {
        execwocallback : async function(queryEst:string, params:oracledb.BindParameters,callback: Function){
            oracledb.getConnection(keys.database, function(err, connection){
                if(err){
                    console.log('Connection error');
                    console.error(err.message);
                    callback();
                }

                connection.execute(queryEst, params,{
                    outFormat: oracledb.OBJECT,
                    autoCommit: true
                },
                function(err,result){
                    if(err){
                        console.log(err.message);
                        doRelease(connection);
                    }
                    else{
                        doRelease(connection);
                    }
                });

                function doRelease(connection:oracledb.Connection){
                    connection.release(
                        function(err){
                            if(err){
                                console.error(err.message);
                            }
                        }
                    );
                }
            })
        },

        exec : async function(queryEst:string,params:oracledb.BindParameters, callback:Function){
            oracledb.getConnection(keys.database,function(err,connection){
                if(err){
                    console.log('Connection error2');
```

Como se manejan las rutas que seran consumidas de la parte de cliente

```
import { Router } from 'express';
import { userController } from '../controllers/userController'
class UserRoutes{
    public router: Router = Router();

    constructor(){
        this.config();
    }

    config():void{
      this.router.get('/', userController.getUser  );
      this.router.post('/login', userController.login  );
      this.router.get('/confirmacion/:id', userController.confirmacion );
      this.router.get('/:id', userController.getOneUser );
      this.router.post('/', userController.create  );
      this.router.post('/sendEmail', userController.emailSend  );
      this.router.post('/update', userController.update);
      this.router.delete('/:id',userController.delete);
      this.router.get('/paises/all', userController.getPaises);
      this.router.post('/mensaje/send',userController.addMensaje);
      this.router.post('/mensaje/enviar',userController.enviarMensaje);
      this.router.get('/mensaje/obtener/:id',userController.getMensajes);
      this.router.get('/mensaje/obtenerChats/:id',userController.getChats);
    }
}

const userRoutes = new UserRoutes();

export default userRoutes.router;
```

Como se trabajan los controladores que haran las consultas a la base de datos



```
import { Request, Response} from 'express';
import pool from '../database';
import * as crypto from 'crypto'; //para incriptar en md5
import nodemailer from 'nodemailer'; //Envia correos, referencias https://nodemailer.com/about/
import jwt from 'jsonwebtoken';
import email from  '../email'

class UserController {
   mensaje:string="Si sale archivos";

   public async getUser(req: Request, res: Response){
        var sql = "SELECT * FROM usuario";

        await pool.db2().exec(sql,[],function(result:any){
            res.json(result);
        });

    }

    public async create(req: Request, res:Response){
        var connection = pool.db2();
        var sql = 'SELECT * FROM usuario WHERE email=:email'
        var obj = req.body;
        var bandera:boolean=false;
        obj.token=jwt.sign(obj.email,obj.nombre);
        obj.pass = crypto.createHash('md5').update(obj.pass).digest("hex"); //Incriptamos la contraseña

        connection.exec(sql,[obj.email],function(result:any){
            if (result.length < 1 ) {
                sql = 'INSERT INTO usuario (nombre,apellido,pass,email,nacimieno,credito,idTipo_U,confirmacion,token,pathI,idPais) VALUES (:nombre,:apellido,:pa

                connection.exec(sql,obj,function(result:any){
                    res.json({text: 'Creado', token: obj.token});
                    bandera=true;
                });
            }else{
                res.json({text: 'Correo ya existe'});
            }

        });

    }
```

Cliente

Para el diseño se utilizo Bootswatch el tema de dark

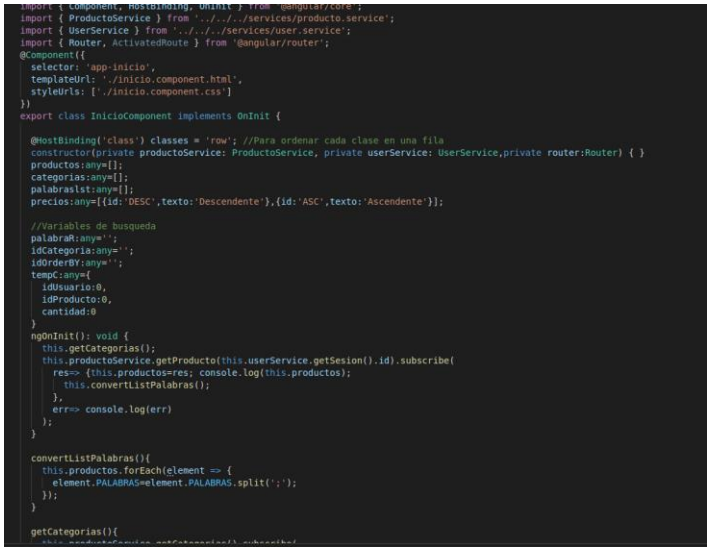
Como se manejan las rutas

```
{
  path: '',
  redirectTo: '/login',
  pathMatch: 'full'
},
{
  path:'login',
  component: LoginComponent,
  canActivate: [AllGuard]
},
{
  path: 'register',
  component: RegisterComponent,
  canActivate: [AllGuard]
},
{
  path: 'confirmacionUser/:id',
  component: ConfirmacionRegistroComponent
},
{
  path: 'cambioContrasenia/:id',
  component: CambioContraseniaComponent
},
{
  path: 'user/home',
  component: HomeComponent,
  canActivate: [UserGuard]
},
{
  path: 'user/newProducto',
  component: NewProductoComponent,
  canActivate: [UserGuard]
},
{
  path: 'user/myProductos',
  component: MisProductosComponent,
  canActivate: [UserGuard]
},
{
  path: 'user/Inicio',
  component: InicioComponent,
  canActivate: [UserGuard]
},
```

Como se manejan los guards para que los usuarios solo se mantengan navegando donde se les hes permitido

```
import { Injectable } from '@angular/core';
import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot, UrlTree,Router} from '@angular/router';
import { Observable } from 'rxjs';
import { UserService } from '../services/user.service'


@Injectable({
  providedIn: 'root'
})
export class AuthGuard implements CanActivate {

  constructor(private userService: UserService,private router:Router){}

  canActivate(
    route: ActivatedRouteSnapshot,
    state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree {
      let sesion = this.userService.getSesion();
      if(sesion.rol== 1){
        return true;
      }
      else{
        this.router.navigate(['/notfound']);
        return false;
      }
  }



}
```

Como se manejan los componentes



```
import { ProductoService } from '../../../services/producto.service';
import { UserService } from '../../../services/user.service';
import { Router, ActivatedRoute } from '@angular/router';
@Component({
  selector: 'app-inicio',
  templateUrl: './inicio.component.html',
  styleUrls: ['./inicio.component.css']
})
export class InicioComponent implements OnInit {

  @HostBinding('class') classes = 'row'; //Para ordenar cada clase en una fila
  constructor(private productoService: ProductoService, private userService: UserService,private router:Router) { }
  productos:any=[];
  categorias:any=[];
  palabraslst:any=[];
  precios:any=[{id:'DESC',texto:'Descendente'},{id:'ASC',texto:'Ascendente'}];

  //Variables de busqueda
  palabraR:any='';
  idCategoria:any='';
  idOrderBY:any='';
  tempC:any={
    idUsuario:0,
    idProducto:0,
    cantidad:0
  }
  ngOnInit(): void {
    this.getCategorias();
    this.productoService.getProducto(this.userService.getSesion().id).subscribe(
      res=> {this.productos=res; console.log(this.productos);
        this.convertListPalabras();
      },
      err=> console.log(err)
    );
  }

  convertListPalabras(){
    this.productos.forEach(element => {
      element.PALABRAS=element.PALABRAS.split(';');
    });
  }

  getCategorias(){
```

Como se manejan los servicios

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { User } from '../models/user';

@Injectable({
  providedIn: 'root'
})
export class UserService {

  constructor(private http: HttpClient) { }
  url:string = "http://192.168.0.8:3009/";

  addUser(user: User){
    return this.http.post(`${this.url}user/`,user);
  }

  getUsers(){
    return this.http.get(`${this.url}user/`);
  }

  login(user: User){
    return this.http.post(`${this.url}user/login`,user);
  }

  setUser(user:any):void{
    let user_string = JSON.stringify(user);
    localStorage.setItem('currentUser',user_string);
  }

  getSesion(){
    let user_string = localStorage.getItem('currentUser');
    return JSON.parse(user_string);
  }

  limpiarSesion(){
    localStorage.removeItem("currentUser");
  }

  confirmacionRegister(token:string){
    return this.http.get(`${this.url}user/confirmacion/${token}`);
  }

  sendEmail(user: User){//Envia email de confirmacion
    return this.http.post(`${this.url}user/sendEmail`,user);
  }
```

Para la app Android se utlizo capacitor