UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
PRACTICAS INICIALES
GRUPO 16

MANUAL TÉCNICO

APLICACIÓN WEB PARA LA CALIFICACIÓN DE DOCENTES DE LA USAC

Anderson Gerardo Zuleta Galdamez 201800500
Fabio Josué Hernández Martinez 201801005
Allan Josué Rafael Morales 201709196
Alejandro José García Hernández 201800939

NOVIEMBRE/2020

UTILIDADES

El proyecto se desarrolló con Angular para el diseño del Front-End, y NodeJS, para diseñar el Back-End, todo esto conectado a una base de datos en MySQL. El lenguaje utilizado para el desarrollo de esté binomio fue Typescript.

Back-End

- Rutas

Publicaciones

```
config() {

    this.router.get('/', publicController.listado);
    this.router.get('/liscurso', publicController.listadoCurso);
    this.router.get('/liscate', publicController.listadoCate);
    this.router.get('/liscatecurso', publicController.listadoCateinCurso);

    this.router.get('/lispublica', publicController.lisPublicaciones);
    this.router.get('/lispublica_on/:idpub', publicController.lisPublicaciones_idp);

    this.router.get('/lispublica_cur/:idCurso', publicController.lisPublicaciones_cur);
    this.router.get('/lispublica_cate/:idCatedratico', publicController.lisPublicaciones_cat);

    this.router.post('/lispublica_nam_cur', publicController.lisPublicaciones_namcur);
    this.router.post('/lispublica_nam_cate', publicController.lisPublicaciones_namecat);

    this.router.post('/',verifyToken,  publicController.createPublicacion);

    this.router.get('/liscom/:idPublicacion', publicController.listadoComentario);
    this.router.post('/insercomen',verifyToken, publicController.createComen);

    this.router.get('/cursospen',verifyToken, publicController.getCursosPen);

    this.router.post('/addCurso',verifyToken, publicController.winCurso);
    this.router.get('/curwin',verifyToken, publicController.getCursosGanados);
    this.router.get('/curwin_id/:iduser', publicController.getCursosGanados_id);
}
```

Usuario

```
config() {

    this.router.get('/', usersController.listusers);
    this.router.post('/', usersController.createUser);
    this.router.post('/Login', usersController.login);
    this.router.post('/Recuperar', usersController.recuperar);

    this.router.get('/getMyProfile', verifyToken, usersController.getProfile);
    this.router.get('/getMyProfile_id/:iduser', usersController.getProfile_id);

    this.router.put('/UpdateUser', verifyToken, usersController.upUser);
}
```

- Conexión a la base

```
export default {

    database: {
        host: 'localhost',
        user: 'root',
        password: '123456789',
        database: 'pracini'
    }

}
```

- Ejecución de la conexión

```
import mysql from 'promise-mysql';

import keys from './dbconfig';

const pool = mysql.createPool(keys.database);

pool.getConnection()
    .then(connection => {
        pool.releaseConnection(connection);
        console.log('DB esta conectada');
    });

export default pool;
```

- Peticiones al servidor

```
config(): void {
    this.app.set('port', process.env.PORT || 3000);

    this.app.use(morgan('dev'));
    this.app.use(cors());
    this.app.use(express.json());
    this.app.use(express.urlencoded({extended: false}));
}
```

- Login

```
public async login(req: Request, res: Response): Promise<any> {
    const { Carnet, constrasenia } = req.body;

    var sql = 'SELECT Carnet from Usuario \
    WHERE Carnet = ? AND constrasenia = ?';

    const result = await pool.query(sql, [Carnet, constrasenia]);
    if (result.length > 0) {

        const tokenid = jwt.sign({idus: Carnet}, 'llave');
        return res.status(200).json({tokenid});
    } else {
        res.status(404).json("Datos incorrectos");
    }

}
```

- Recuperar contraseña

```
public async recuperar(req: Request, res: Response): Promise<any> {
  const { Carnet, correo, constrasenia } = req.body;

  var sql = 'SELECT Carnet from Usuario \
  WHERE Carnet = ? AND correo = ?';

  const result = await pool.query(sql, [Carnet, correo]);

  if (result.length > 0) {

      /*si es correcto entonces actulizo contrasña*/
      var sql_up = "UPDATE Usuario SET constrasenia = ?\
      WHERE Carnet = ?";

      const result_up = await pool.query(sql_up, [constrasenia, Carnet] )
      const tokenid = jwt.sign({idus: Carnet}, 'llave');

      return res.status(200).json({tokenid});
  } else {
      res.status(404).json("Datos incorrectos");
  }
```

- Perfil de usuario

```
public async getProfile_id(req: Request, res: Response): Promise<void> {
  const { iduser } = req.params;
  var Carnet  = iduser
  var sql = "SELECT Carnet, Nombres, Apellidos, constrasenia, correo\
  FROM Usuario WHERE Carnet = ?"
    try {
        const result = await pool.query(sql, [Carnet]);
        res.status(200).json(result);
      } catch (err) {
        res.status(404).json({ status: -1, error: err.sqlMessage });
      }
}
```

- Crear usuario

```
public async createUser(req: Request, res: Response): Promise<void> {
    const { Carnet, Nombres, Apellidos, constrasenia, correo } = req.body;
    var sql = "INSERT INTO Usuario (Carnet, Nombres, Apellidos, constrasenia, correo) VALUES (?, ?, ?, ?, ?)";


try {
    const result = await base_con.query(sql, [Carnet, Nombres, Apellidos, constrasenia, correo] );
    res.status(200).json('usuario ingresado' );
  } catch (err) {
    res.status(404).json(err.sqlMessage);
  }

}
```

- Comentarios publicados

```
public async listadoComentario(req: Request, res: Response): Promise<any> {
    const { idPublicacion } = req.params;
   // console.log("idPublicacion", idPublicacion)
   var sql = "SELECT comentario.idComentario, comentario.Comentario, comentario.idPublicacion, comentario.Carnet,\
   CONCAT(Usuario.Nombres, ' ' ,Usuario.Apellidos ) as alumno\
   FROM comentario, Usuario\
   WHERE comentario.Carnet = Usuario.Carnet AND\
   idPublicacion = ?"
    try {
        const result = await pool.query(sql, [idPublicacion]);
        //res.json({ status: 1, message: 'usuario ingresado' });
        res.json(result);
      } catch (err) {
        res.status(404).json({ status: -1, error: err.sqlMessage });
        // handle errors here
      }
}
```

Modelo Entidad Relacional