

Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de ingeniería en Ciencias y Sistemas  
Laboratorio de Manejo e Implementación de Archivos

## **Manual Técnico**

Luis Fernando Velasquez Zacarias  
201807266  
noviembre de 2020



## SQL ORACLE 18c

Para la consistencia de datos se utiliza un servidor de base de datos Oracle en la versión 18c. Se montó sobre un contenedor de docker local.

```
@5da91e42ed1f:/

System altered.

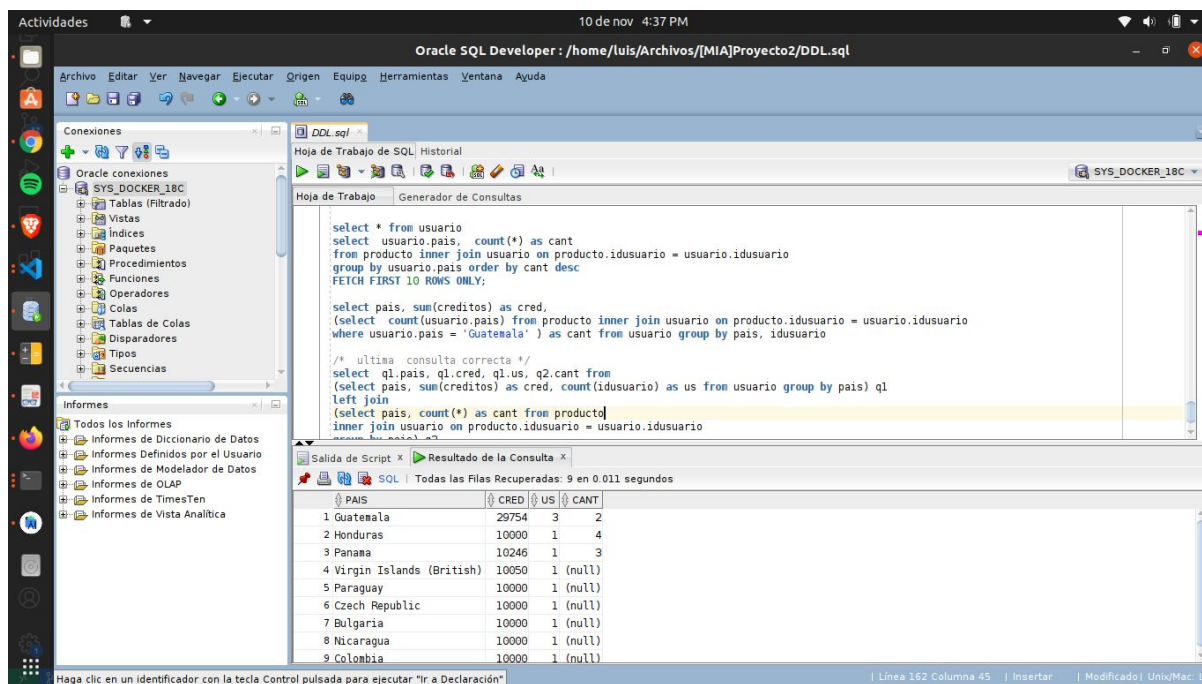
SQL> alter pluggable database PDB18C open
*
ERROR at line 1:
ORA-65019: pluggable database PDB18C already open

SQL>
NAME          OPEN_MODE
-----
ORCL18        READ WRITE

SQL>
CON_ID  CON_NAME                                OPEN MODE  RESTRICTED
-----
2 PDB$SEED                                READ ONLY  NO
3 PDB18C                                  READ WRITE NO

SQL> Disconnected from Oracle Database 18c Enterprise Edition Release 18.0.0.0.0
- Production
Version 18.3.0.0.0
Cleaning up
DataBase Installed!!!
[oracle@5da91e42ed1f /]$
```

En este se definió el modelo relacional que resolvía las necesidades de la aplicación





## NODE JS

Para la comunicación con la base de datos se montó un servidor de nodejs, para recibir las peticiones HTTP, a través de una API REST

```
[nodemon] starting `node /src/index.js src/index.js`  
Server on Port 3000  
GET /getBitacora/ 304 675.425 ms - -  
GET /getrep6/ 200 213.699 ms - 280  
GET /getrep4a/ 200 221.115 ms - 667  
GET /getrep1/ 200 223.787 ms - 162  
GET /getrep4b/ 200 222.080 ms - 666  
GET /getrep5/ 200 221.103 ms - 290  
GET /getrep7/ 200 71.325 ms - 617  
GET /getBitacora/ 304 111.762 ms - -  
GET /getrep4a/ 304 97.465 ms - -  
GET /getrep1/ 304 108.016 ms - -
```

se utilizan diferentes paquetes de npm para monitorear las peticiones y errores, como morgan, cors, express

```
const express = require('express');  
const morgan = require('morgan');  
const cors = require('cors');  
const app = express();  
  
//imports  
const usuarios = require('./routes/usuarios');  
const email = require('./routes/email');  
const categorias = require('./routes/categorias');  
const productos = require('./routes/productos');  
const carrito = require('./routes/carrito');  
const comentarios = require('./routes/comentarios');  
const denuncias = require('./routes/denuncias');  
const reportes = require('./routes/reportes');  
  
//settings  
app.set('port', 3000);  
  
//middlewares  
app.use(morgan('dev'));  
app.use(express.json());  
app.use(express.urlencoded({ extended: false }));  
app.use(cors());  
  
//routes  
app.use(usuarios);  
app.use(email);  
app.use(categorias);
```

Este servidor corre en el puerto 3000, realiza las consultas a la base de datos



```
router.put("/passusuario", async (req, res) => {
  const { pass, idusuario } = req.body;

  sql = "update usuario set pass=:pass where idusuario=:idusuario";
  await BD.Open(sql, [pass, idusuario], true);

  res.status(200).json({
    "pass": pass,
    "idusuario": idusuario
  })
})

router.put("/pagousuario", async (req, res) => {
  const { creditos, idusuario } = req.body;

  sql = "update usuario set creditos=(creditos + (select creditos from usuario where idusuario=:idusuario))\
  where idusuario=:idusuario";
  await BD.Open(sql, [creditos, idusuario, idusuario], true);

  res.status(200).json({
    "creditos": creditos,
    "idusuario": idusuario
  })
})
```





## ANGULAR 9

para el cliente web se utilizó el framework de google angular, para creacion de aplicaciones dinámicas de una sola pagina.

```
<!-- Import Google Icon Font -->
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
<meta charset="utf-8">
<title>Gtsales</title>
<base href="/">
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
      integrity="sha384-TX8t27EcRE3e/ihU7zmQxvncDAy5uIKz4rEkGIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="an">
<link rel="icon" type="image/x-icon" href="favicon.ico">
</head>

<body>
  <app-root></app-root>
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
    integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+0GpamoFVy38MVBnE+IbbVYUew+0rCXaRkfj"
    crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGG0n+Y7RsweNrtN/tE3MoK7ZeZDyx"
    crossorigin="anonymous"></script>
```

Ese al estar basado en nodejs, tiene módulos, que se declaran en el módulo principal, y componentes que son las fracciones de código variable que se muestra en el navegador.

```
reportes.component.html  TS reportes.component.ts  TS app.module.ts X  TS denunciadas.component.ts  TS producto.service.ts

src > app > TS app.module.ts > ...
22 import { DenunciaComponent } from './components/usuarios/denuncia/denuncia.component';
23 import { DenunciadasComponent } from './components/administrador/denunciadas/denunciadas.component';
24 import { ReportesComponent } from './components/administrador/reportes/reportes.component';
25
26 @NgModule({
27   declarations: [
28     AppComponent,
29     UsuariosComponent,
30     LoginComponent,
31     RegistroComponent,
32     RecuperarComponent,
33     NavusuariosComponent,
34     PerfilComponent,
35     AdministradorComponent,
36     NavadministradorComponent,
37     CategoriasComponent,
38     ProductosComponent,
39     ComparComponent,
40     ComentariosComponent,
41     DenunciaComponent,
42     DenunciadasComponent,
43     ReportesComponent
44   ],
45   imports: [
46     BrowserModule,
47     AppRoutingModule,
48     HttpClientModule,
49     FormsModule,
50     ReactiveFormsModule,
51   ],
52   providers: [],
53   bootstrap: [AppComponent]
```

