Introducción

Para el desarrollo de la solución se utilizó el lenguaje de programación Go y el gestor de base de datos MySQL en el cual se estructuró una base de datos remota con tres tablas diferentes:

Tabla Usuarios:

ID_user	Nombre_User	Ducument	Telefono	Sexo	Tipo_Doc_ID	Rol_User
9	Andres	1116276052	3113908205	1	1	2
10	Prueba	1116276052	3113908205	0	2	2
11	Diana C	123456	88888	0	1	2

Tabla Tiplo_Documento:

ID_Documento	Tipo_Documento
1	CC
2	NIT

Campos permitidos: “CC”, “NIT”, “TI”, “CE”.

Tabla Tipo_Rol:

ID_Rol	Tipo_Rol
1	Super_Admin
2	Admin

Campos permitidos: “Super_Admin”, “Admin”, “Cajero”, “Gerente”.

Recomendaciones

Para el correcto funcionamiento del api deben ser instaladas las dependencias gorilla/mux y MySQL

- go get -u github.com/gorilla/mux
- github.com/go-sql-driver/mysql"

El api corre en el puerto **8080.**

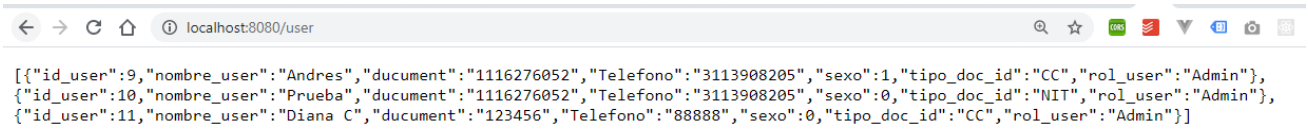
Puede acceder a la solución en la siguiente dirección **http://localhost:8080/user**

La petición post debe tener la siguiente estructura para el caso de peticiones PUT y POST:

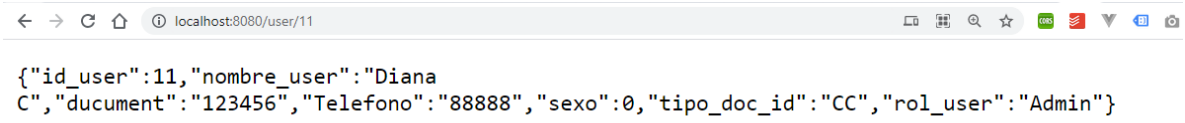
- {"nombre_user":"Diana","ducument":"123456","Telefono":"88888","sexo":0,"tipo_doc_id":"CC","rol_user":"Admin"}

Casos de prueba

Caso de prueba 1 (GET):



Caso de prueba 2 (GET/{11}):



Caso de prueba 3 (POST):

- **Entrada:**

```
{
"nombre_user":"Fabian",
"ducument":"111522486",
"Telefono":"311311311",
"sexo":1,
"tipo_doc_id":"CC",
"rol_user":"Admin"
}
```

- **Salida:**

```
{
“id_user”: 13
"nombre_user":"Fabian",
"ducument":"111522486",
"Telefono":"311311311",
"sexo":1,
"tipo_doc_id":1,
"rol_user":2
}
```

Caso de prueba 4 (PUT/{13}):

- **Entrada:**
  {"nombre_user":"Mauricio",
  "ducument":"123456",
  "Telefono":"88888",
  "sexo":1,
  "tipo_doc_id":"CC",
  "rol_user":"Admin"}

- **Salida:** “Se actualizó el correctamente el usuario: 13”

Caso de prueba 5 (DELETE/{10}):

- **Salida: “**El usuario se eliminó con éxito**”**