FUNDAMENTOS DE PROGRAMACIÓN

Grupo 8

Scrum Master

Diego Fabián Sánchez Villamil

Development Team

Giezi A. Diaz. Morales
Jhon Walter Riveros
Michael Cortes

unab
Universidad Autónoma de Bucaramanga

MINTIC

CONTENIDO

SPRINT PLANNING 3
DAILY MEETING 4
ACT-01 4
ACT-02 5
SPRINT 5 6
Product Increment 7
CLIENTES 7
crearClientes 7
eliminarClientes 10
CLASES 11
crearClases 11
consultarClases 16
eliminarClases 16
INSTRUCTORES 17
crearInstructores 17
consultarInstructores 22
eliminarInstructores 22
HORARIOS 23
crearHorarios 23
ConsultarHorarios 26
eliminarHorarios 27
LOGIN 28
consultaLogin 28
TALLER1 30
Despliegue aplicación en el servidor 34
SPRINT 6 37
Product Increment 40
CLIENTES 40

SPRINT PLANNING

Fecha: 04/10/2021

Horario: 8:15 pm a 8:30 am

Recursos:

Enlace: **https://drive.google.com/file/d/1z4FJNjDyf7qkeUu9GlOpxfOCH7Q7oCrz/view?usp=sharing**

GitHub: https://github.com/DI3GOS/G8_Proyecto

Trello: https://trello.com/b/lGSvWpi6/g8proyectofinal

Descripción: Se realiza la asignación de las actividades del sprint 5, En donde se deben implementar los servicios en rest api usando bootstart

Participantes:

1. Scrum Master
   - Diego Fabián Sánchez Villamil
2. Development Team
   - Giezi A. Diaz. Morales
   - Jhon Walter Riveros
   - Michael Cortes

Para desarrollar el Sprint 5, se desarrollaron tres (2) Daily Meeting, como se muestra a continuación:

DAILY MEETING

ACT-01
Fecha: 11/10/2021

Horario: 8:15 pm a 8:30 am

Recursos:

Enlace: https://drive.google.com/file/d/1z4FJNjDyf7qkeUu9GlOpxfOCH7Q7oCrz/view?usp=sharing

GitHub: https://github.com/DI3GOS/G8_Proyecto

Trello: https://trello.com/b/lGSvWpi6/g8proyectofinal

Descripción: revisión de las asignaciones de los rest api. Cliente, clases e instructores
Participantes:

1. Scrum Master
   - Diego Fabián Sánchez Villamil
2. Development Team
   - Giezi A. Diaz. Morales
   - Jhon Walter Riveros
   - Michael Cortes

ACT-02
Fecha: 14/10/2021

Horario: 8:15 pm a 8:30 am

Recursos:

Enlace: https://drive.google.com/file/d/1mg9X3g7voMLO3x_Tl-S5-vPBg_gasinw/view?usp=sharing

GitHub: https://github.com/DI3GOS/G8_Proyecto

Trello: https://trello.com/b/lGSvWpi6/g8proyectofinal

Descripción: revisión de las asignaciones de los rest api. Horarios y login
Participantes:

1. Scrum Master
   - Diego Fabián Sánchez Villamil
2. Development Team
   - Giezi A. Diaz. Morales
   - Jhon Walter Riveros
   - Michael Cortes

SPRINT 5

El cuarto ***Sprint*** que tiene una duración de 2 semana, en mi caso lo desarrollaré así: realicé la continuación de los archivos HTML con CSS creados en el Sprint 3, se agregaron códigos JavaScript, utilizando el código del FrameWork Bootstrap y JS.

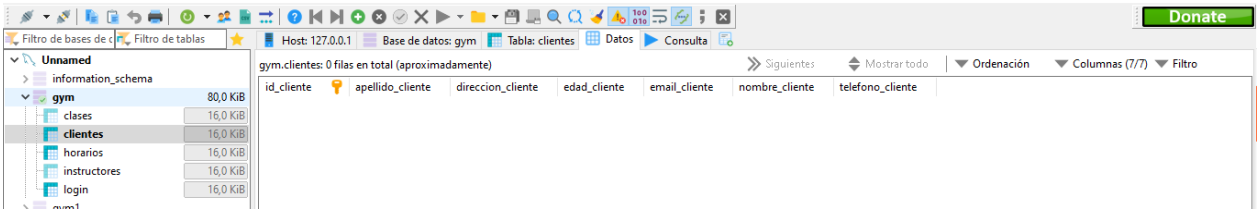
	No. Historias de Usuario	***Nombre de la Historia de Usuario***	***Como <tipo de usuario>***	***Quiero <realizar alguna tarea>***	***Para <completar mis tareas a tiempo>***	***Criterios de Aceptación***
	ACT-20	***Creación Web services clientes***	Diego Sanchez	Crear servicios Bootstrap clientes	Para crear y probar los web services de clientes	Se deben implementar un web service para exponer las operaciones crearCliente, obtenerClientes y eliminarCliente.
	ACT-21	***Creación Web services clases***	Michael Cortes	Crear servicios Bootstrap ***clases***	Para crear y probar los web services de ***clases***	Se deben implementar un web service para exponer las operaciones crearClases, obtenerClases y eliminarClases.
	ACT-21	***Creación Web services instructores***	Jhon Walter Riveros	Crear servicios Bootstrap ***instructores***	Para crear y probar los web services de ***instructores***	Se deben implementar un web service para exponer las operaciones crearInstructores, obtenerInstructores y eliminarInstructores.
	ACT-22	***Creación Web services horarios***	Giezi A. Diaz. Morales	Crear servicios Bootstrap ***horarios***	Para crear y probar los web services de ***horarios***	Se deben implementar un web service para exponer las operaciones crearHorarios, obtenerHorarios y eliminarHorarios.
	ACT-22	***Creación Web services horarios***	Diego Sanchez	Crear servicios Bootstrap ***lolgin***	Para crear y probar los web services de ***login***	Se deben implementar un web service para exponer las operaciones consultarRoles

Product Increment

CLIENTES

crearClientes

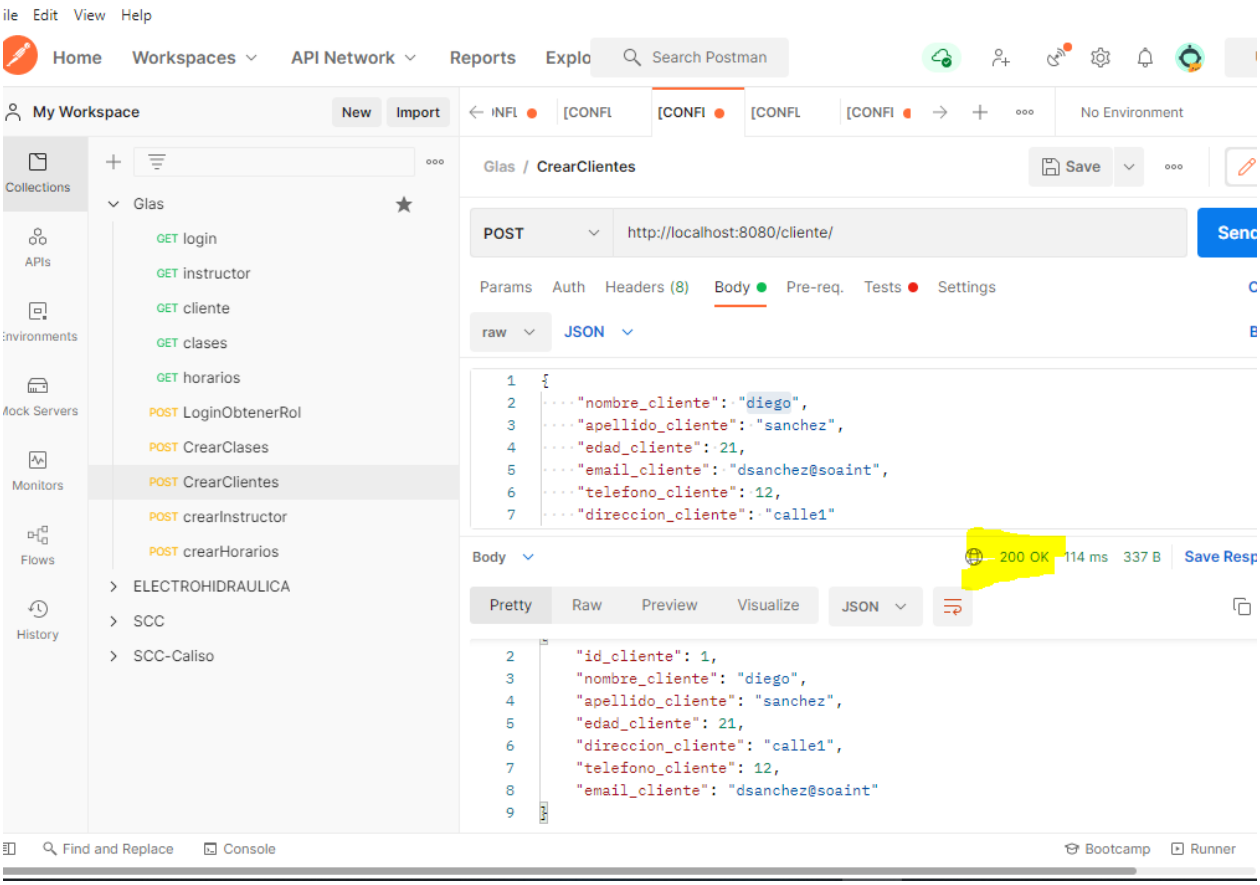
Base de datos Vacia

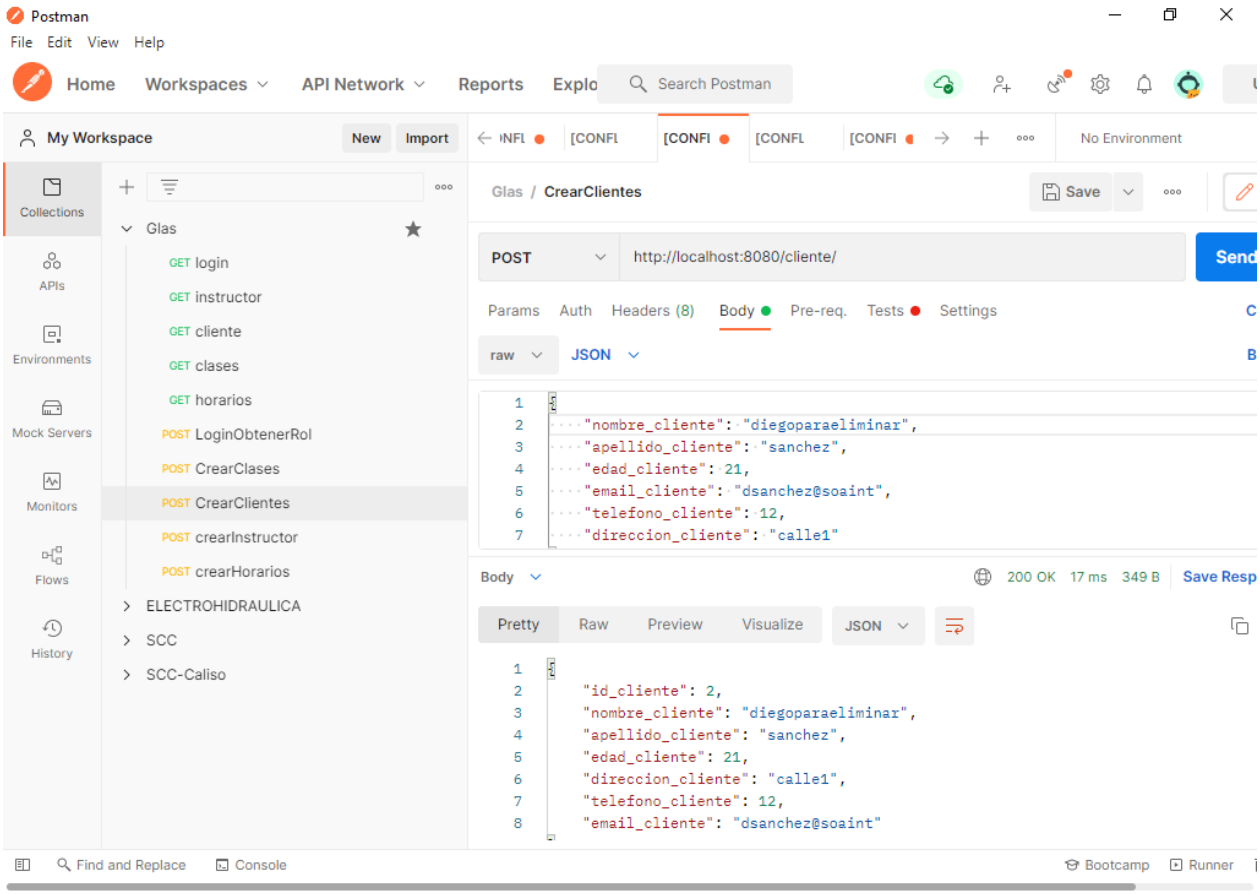


Creación de 2 registros por el postman

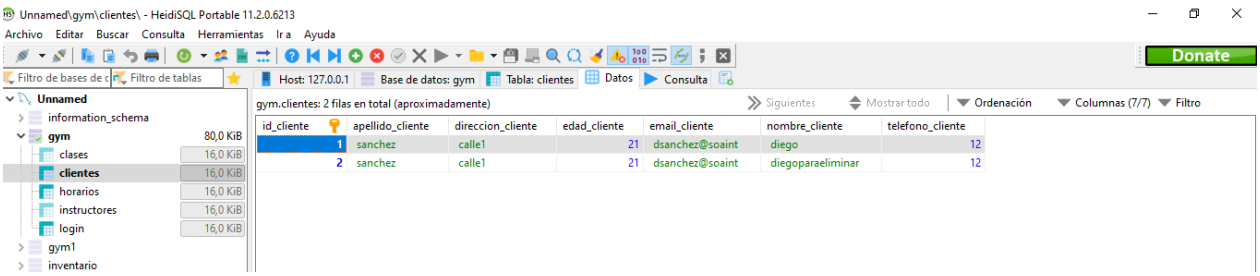
JSON

```
{
    "nombre_cliente": "diego",
    "apellido_cliente": "sanchez",
    "edad_cliente": 21,
    "email_cliente": "dsanchez@soaint",
    "telefono_cliente": 12,
    "direccion_cliente": "calle1"
}
```


ile Edit View Help
Home
Workspaces
API Network
Reports
Explo
Search Postman
My Workspace
New
Import
No Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
Glas
GET login
GET instructor
GET cliente
GET clases
GET horarios
POST LoginObtenerRol
POST CrearClases
POST CrearClientes
POST crearInstructor
POST crearHorarios
ELECTROHIDRAULICA
SCC
SCC-Caliso
Glas / CrearClientes
Save
POST
http://localhost:8080/cliente/
Send
Params
Auth
Headers (8)
Body
Pre-req.
Tests
Settings
raw
JSON
{
"nombre_cliente": "diego",
"apellido_cliente": "sanchez",
"edad_cliente": 21,
"email_cliente": "dsanchez@soaint",
"telefono_cliente": 12,
"direccion_cliente": "calle1"
Body
200 OK
114 ms
337 B
Save Resp
Pretty
Raw
Preview
Visualize
JSON
"id_cliente": 1,
"nombre_cliente": "diego",
"apellido_cliente": "sanchez",
"edad_cliente": 21,
"direccion_cliente": "calle1",
"telefono_cliente": 12,
"email_cliente": "dsanchez@soaint"
Find and Replace
Console
Bootcamp
Runner

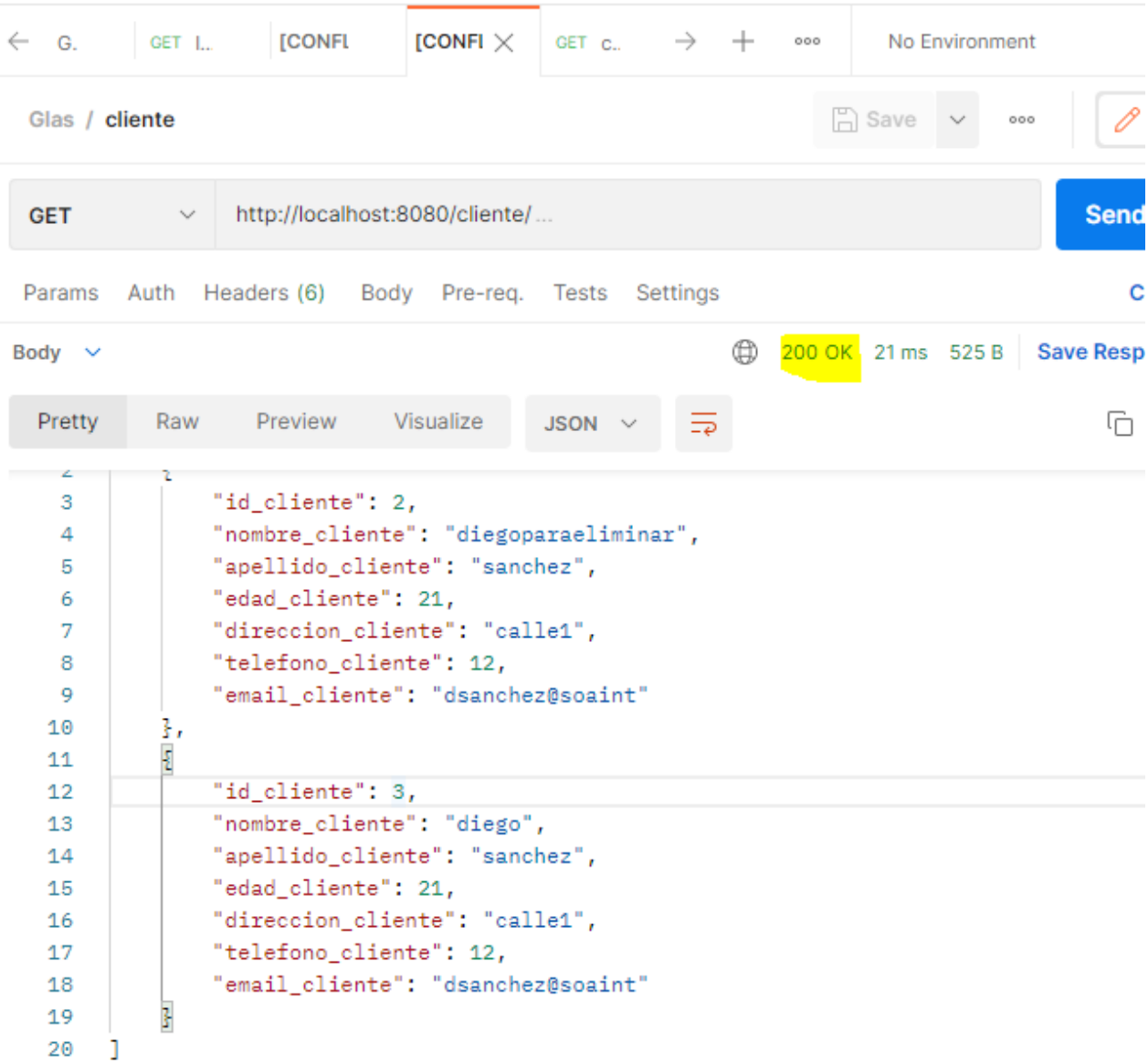


Base de datos con los 2 registros ingresados



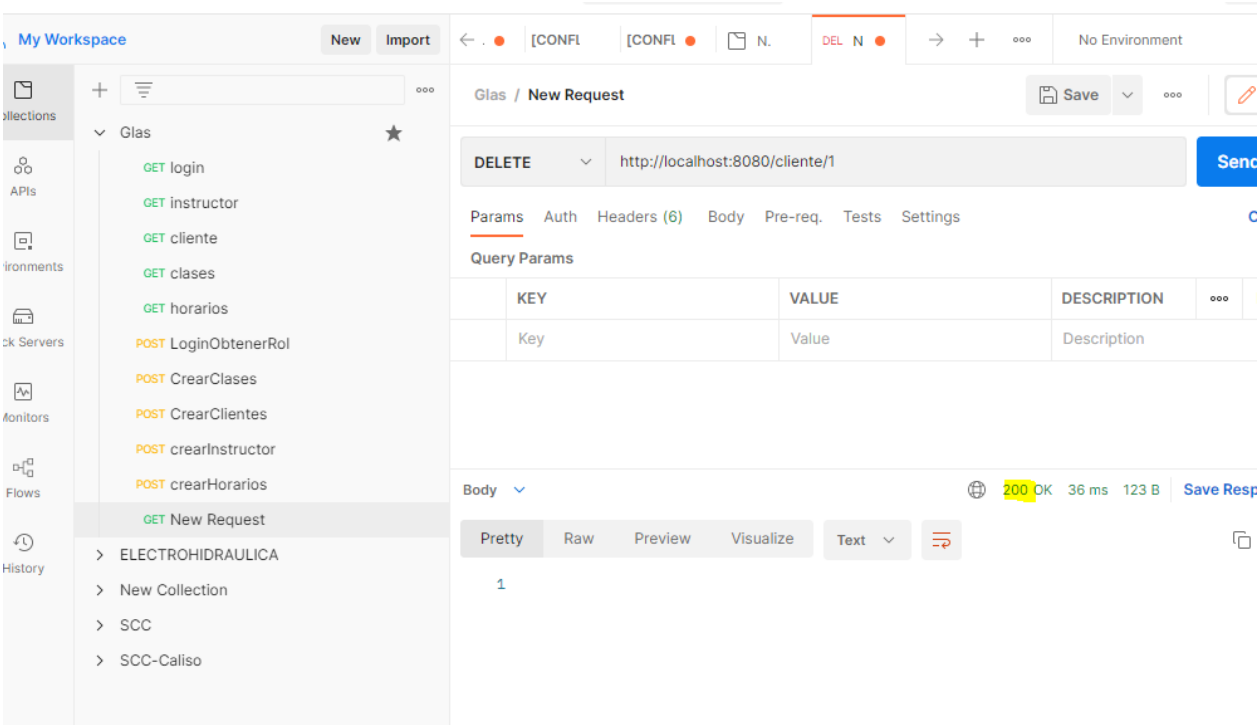
consultarClientes

consulta de registros por el postman

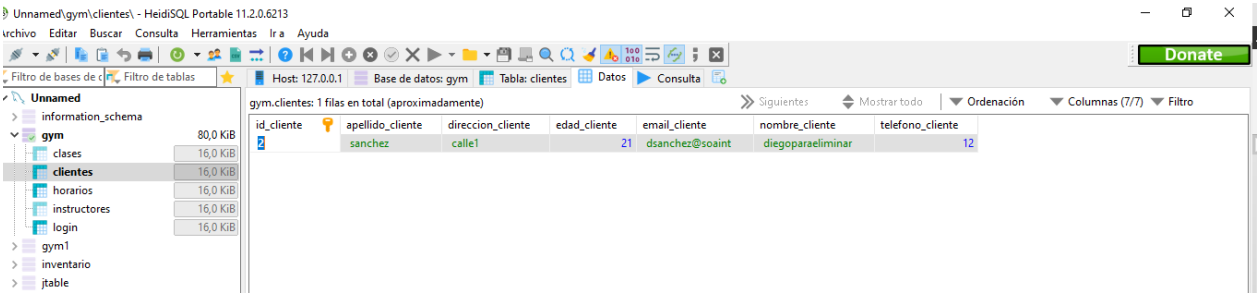


eliminarClientes

Eliminar el registro con código 1 por el postman



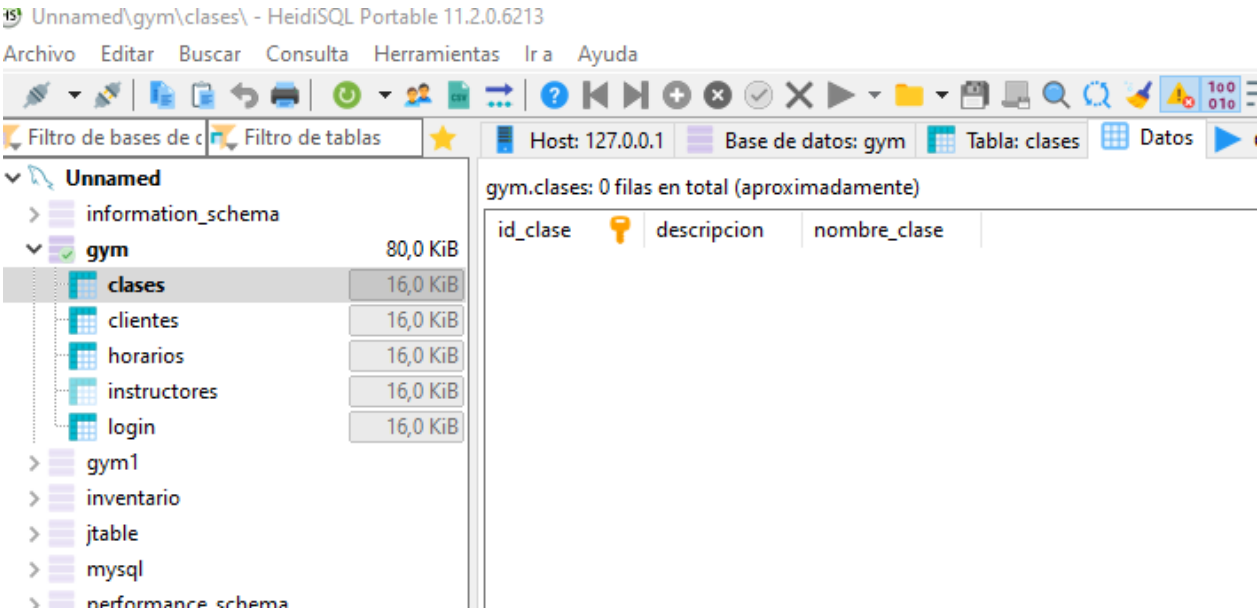
Base de datos sin el registro con código 1



CLASES

crearClases

Base de datos sin datos



Crear Clases por el postman+

JSON

```
{
    "nombre_clase": "Spinning",
    "descripcion": "Realizar Spinning"
}
```

POST http://localhost:8080/clases/ ...

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

```
1 {
2     "nombre_clase": "Aerobic",
3     "descripcion": "Realizar Aerobics"
4 }
```

Body 200 OK 19 ms 237 B Save R

Pretty Raw Preview Visualize JSON

```
1 {
2     "id_clase": 1,
3     "nombre_clase": "Aerobic",
4     "descripcion": "Realizar Aerobics"
5 }
```

Glas / **CrearClases**

Save

POST http://localhost:8080/clases/ ...

Se

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

```
1 {
2     "nombre_clase": "Zumba",
3     "descripcion": "Realizar Zumba"
4 }
```

Body 200 OK 11 ms 232 B Save Re

Pretty Raw Preview Visualize JSON

```
1 {
2     "id_clase": 2,
3     "nombre_clase": "Zumba",
4     "descripcion": "Realizar Zumba"
5 }
```

Glas / CrearClases

Save

POST http://localhost:8080/clases/ ...

Send

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

```
{
    "nombre_clase": "Spinning",
    "descripcion": "Realizar Spinning"
}
```

Body 200 OK 8 ms 238 B Save Resp

Pretty Raw Preview Visualize JSON

```
{
    "id_clase": 3,
    "nombre_clase": "Spinning",
    "descripcion": "Realizar Spinning"
}
```

Registros en la base de datos

Archivo Editar Buscar Consulta Herramientas Ir a Ayuda

Filtro de bases de c Filtro de tablas Host: 127.0.0.1 Base de datos: gym Tabla: clases Datos

Unnamed
- information_schema
- gym 80,0 KiB
  - clases 16,0 KiB
  - clientes 16,0 KiB
  - horarios 16,0 KiB
  - instructores 16,0 KiB
  - login 16,0 KiB
- gym1
- inventario
- jtable
- mysql
- performance_schema
- sia

gym.clases: 3 filas en total (aproximadamente)

id_clase	descripcion	nombre_clase
1	Realizar Aerobics	Aerobic
2	Realizar Zumba	Zumba
3	Realizar Spinning	Spinning

consultarClases

Consultar el servicio para ver las clases

Glas / clases

Save

GET http://localhost:8080/clases/

Se

Params Auth Headers (6) Body Pre-req. Tests Settings

Body 200 OK 18 ms 383 B Save Re

Pretty Raw Preview Visualize JSON


```
[
    {
        "id_clase": 1,
        "nombre_clase": "Aerobic",
        "descripcion": "Realizar Aerobics"
    },
    {
        "id_clase": 2,
        "nombre_clase": "Zumba",
        "descripcion": "Realizar Zumba"
    },
    {
        "id_clase": 3,
        "nombre_clase": "Spinning",
        "descripcion": "Realizar Spinning"
    }
]
```


eliminarClases

Se elimina la clase con código 2

Glas / New Request

Save

DELETE http://localhost:8080/clases/2 Sen

Params Auth Headers (6) Body Pre-req. Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body 200 OK 15 ms 123 B Save Res

Pretty Raw Preview Visualize Text

1

Base de datos sin el registro

Filtro de bases de c Filtro de tablas Host: 127.0.0.1 Base de datos: gym Tabla: clases Datos Consulta

Unnamed
- information_schema
- gym 80,0 KiB
  - clases 16,0 KiB
  - clientes 16,0 KiB
  - horarios 16,0 KiB
  - instructores 16,0 KiB
  - login 16,0 KiB
- gym1
- inventario
- jtable
- mysql
- performance_schema
- sia 144,0 KiB

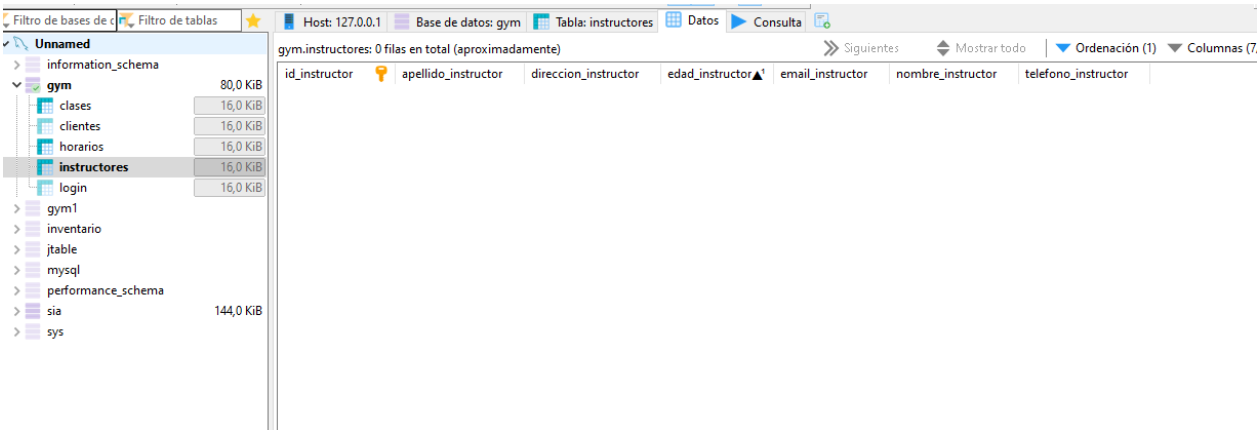
gym.clases: 2 filas en total (aproximadamente)

id_clase	descripcion	nombre_clase
1	Realizar Aerobics	Aerobic
3	Realizar Spinning	Spinning

INSTRUCTORES

crearInstructores

Base de datos sin datos



Ejecución del servicio para crear clientes por el postman

JSON

```
{
    "nombre_instructor": "instructor3",
    "apellido_instructor": "apellido3",
    "edad_instructor": 1,
    "direccion_instructor": "12",
    "telefono_instructor": 1,
    "email_instructor": "11"
}
```

[CONFI [CONFL N. DEL N No Environment

Glas / **crearInstructor** Save

POST http://localhost:8080/instructor/...

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

```
{
    "nombre_instructor": "instructor1",
    "apellido_instructor": "1",
    "edad_instructor": 1,
    "direccion_instructor": "12",
    "telefono_instructor": 1,
    "email_instructor": "11"
}
```

Body 200 OK 18 ms 339 B Save R

Pretty Raw Preview Visualize JSON

```
{
    "id_instructor": 1,
    "nombre_instructor": "instructor1",
    "apellido_instructor": "1",
    "edad_instructor": 1,
    "direccion_instructor": "12"
```

Glas / **crearInstructor**

Save

POST http://localhost:8080/instructor/...

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

```
{
    "nombre_instructor": "instructor2",
    "apellido_instructor": "apellido2",
    "edad_instructor": 1,
    "direccion_instructor": "12",
    "telefono_instructor": 1,
    "email_instructor": "11"
}
```

Body 200 OK 11 ms 347 B Sav

Pretty Raw Preview Visualize JSON

```
{
    "id_instructor": 2,
    "nombre_instructor": "instructor2",
    "apellido_instructor": "apellido2",
    "edad_instructor": 1,
    "direccion_instructor": "12"
```

Glas / **crearInstructor**

Save

POST http://localhost:8080/instructor/...

Send

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

```
{
    "nombre_instructor": "instructor3",
    "apellido_instructor": "apellido3",
    "edad_instructor": 1,
    "direccion_instructor": "12",
    "telefono_instructor": 1,
    "email_instructor": "11"
}
```

Body 200 OK 9 ms 347 B Save Resp

Pretty Raw Preview Visualize JSON

```
{
    "id_instructor": 3,
    "nombre_instructor": "instructor3",
    "apellido_instructor": "apellido3",
    "edad_instructor": 1,
    "direccion_instructor": "12"
```

Bootcamp Runner

Consulta de base de datos con la información ingresada

Host: 127.0.0.1 | Base de datos: gym | Tabla: instructores | Datos | Consulta

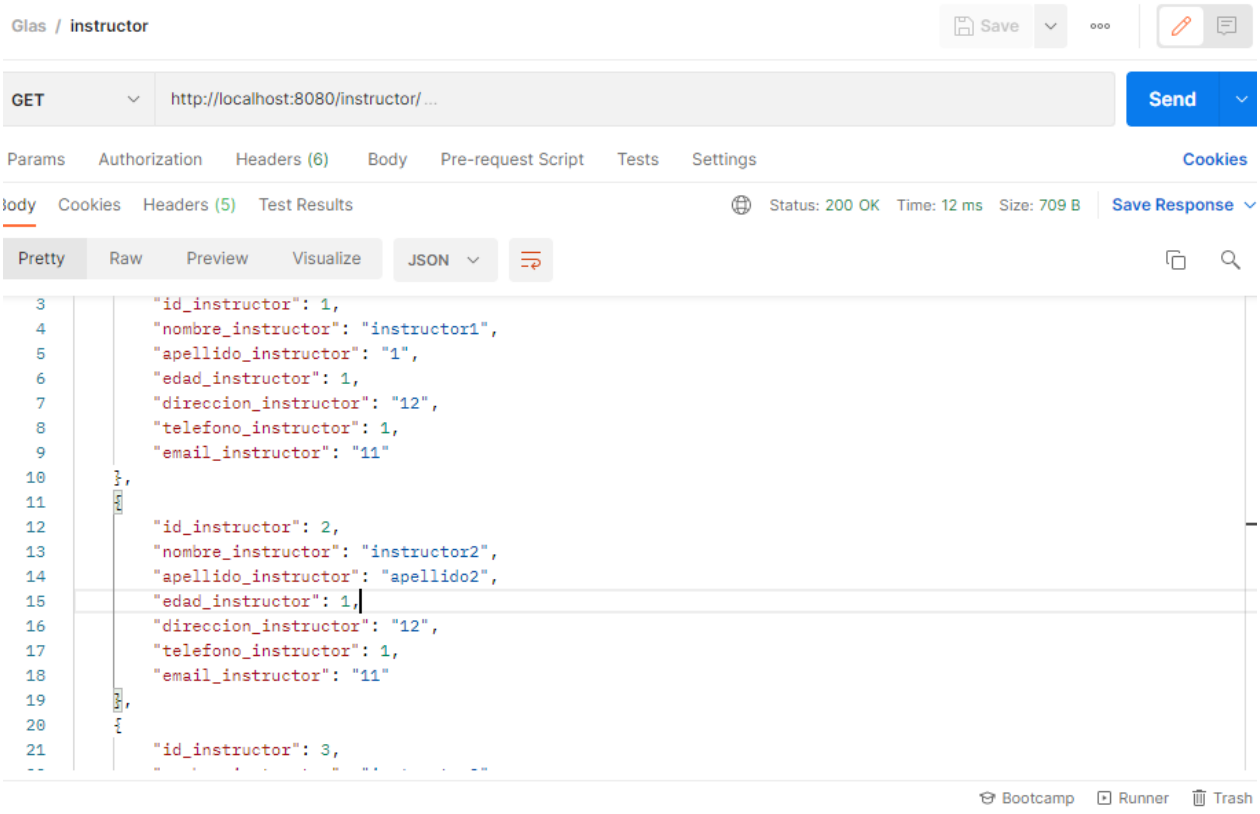
gym.instructores: 3 filas en total (aproximadamente)

id_instructor	apellido_instructor	direccion_instructor	edad_instructor	email_instructor	nombre_instructor	telefono_instructor
1	1	12	1	11	instructor1	1
2	apellido2	12	1	11	instructor2	1
3	apellido3	12	1	11	instructor3	1

Unnamed
- information_schema
- gym 80,0 KiB
  - clases 16,0 KiB
  - clientes 16,0 KiB
  - horarios 16,0 KiB
  - instructores 16,0 KiB
  - login 16,0 KiB
- gym1
- inventario
- jtable
- mysql
- performance_schema
- sia 144,0 KiB

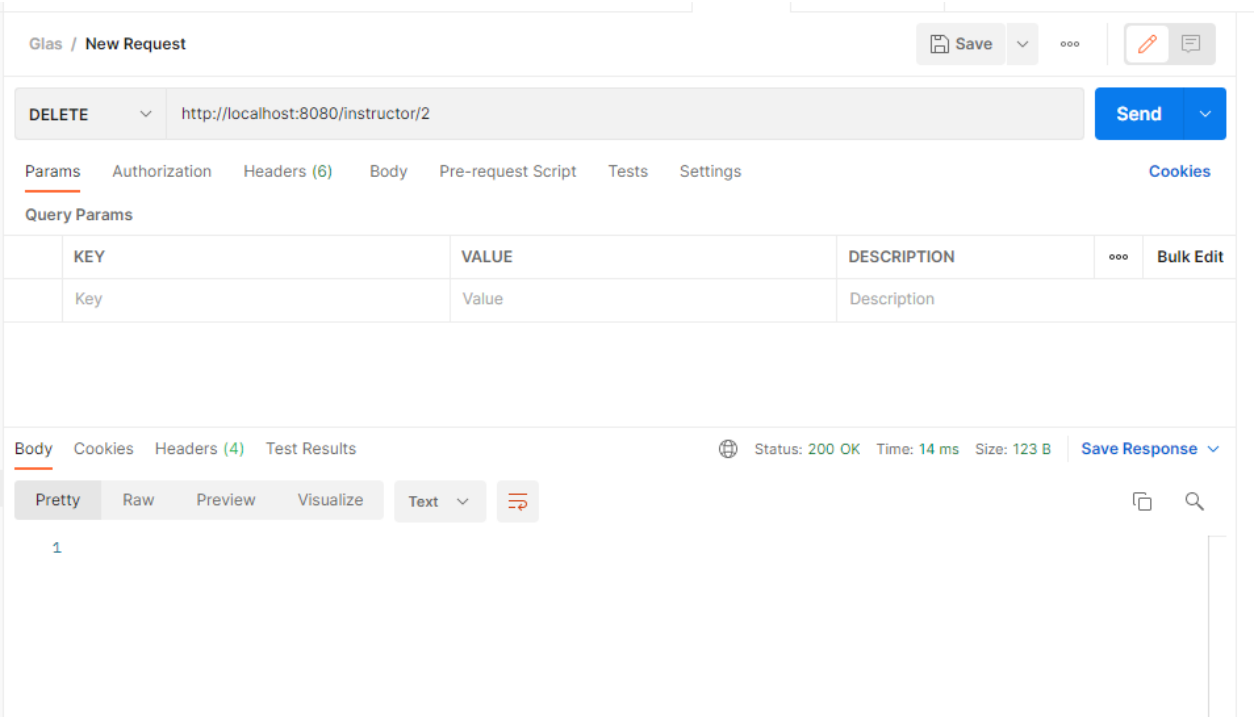
consultarInstructores

consultar instructores por el postman

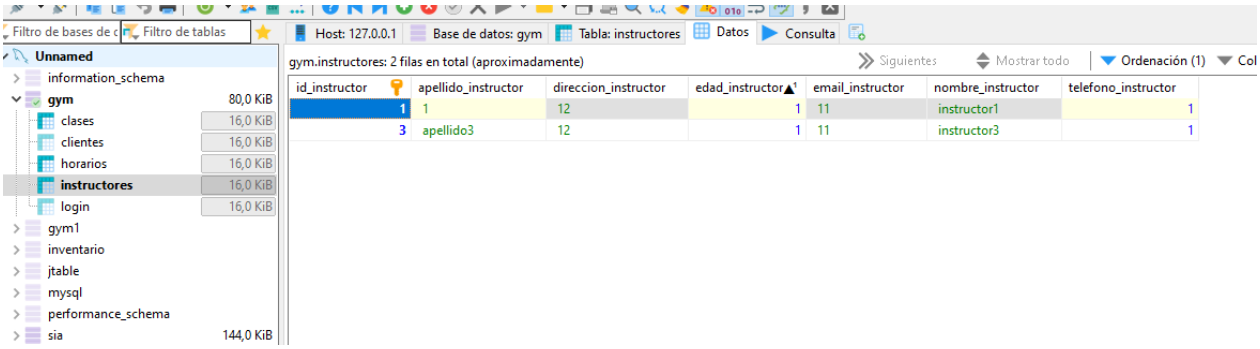


eliminarInstructores

Eliminar el instructor con código 2 por el postman



Consulta de la base de datos después de eliminar el instructor con código 2



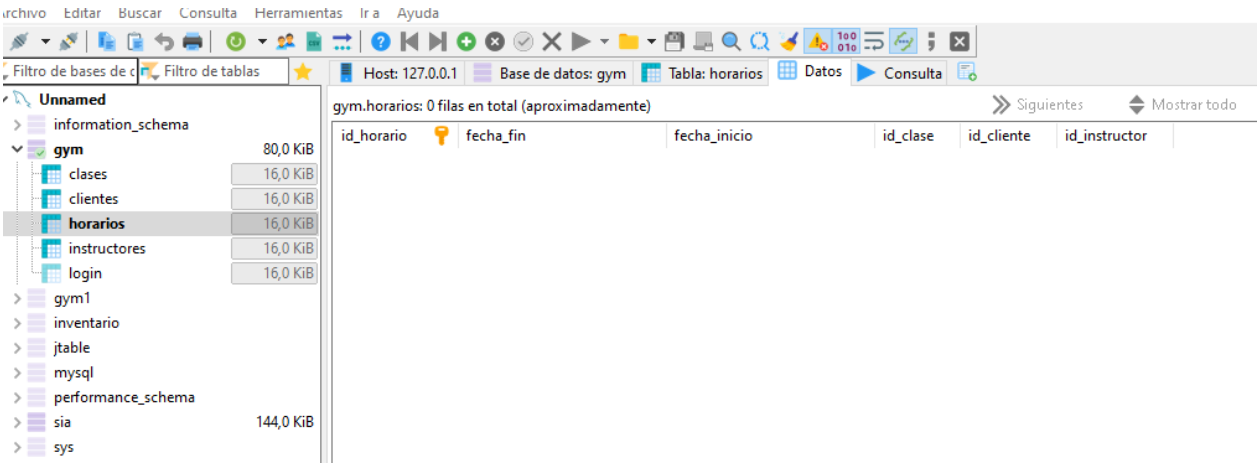
HORARIOS

crearHorarios

Consulta a la base de datos sin registros

Id clase 1, cliente 3, instructor 3

Id clase 2, cliente 2 instructor1



JSON


```
{
    "fecha_fin": "2021-10-30T08:00",
    "fecha_inicio": "2021-10-30T10:00",
    "id_cliente": 3,
    "id_clase": 1,
    "id_instructor": 3
}
```

INFI [CONFI [CONFI N. DEL N No Environment

Glas / **crearHorarios** Save

POST http://localhost:8080/horarios/ Sen

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

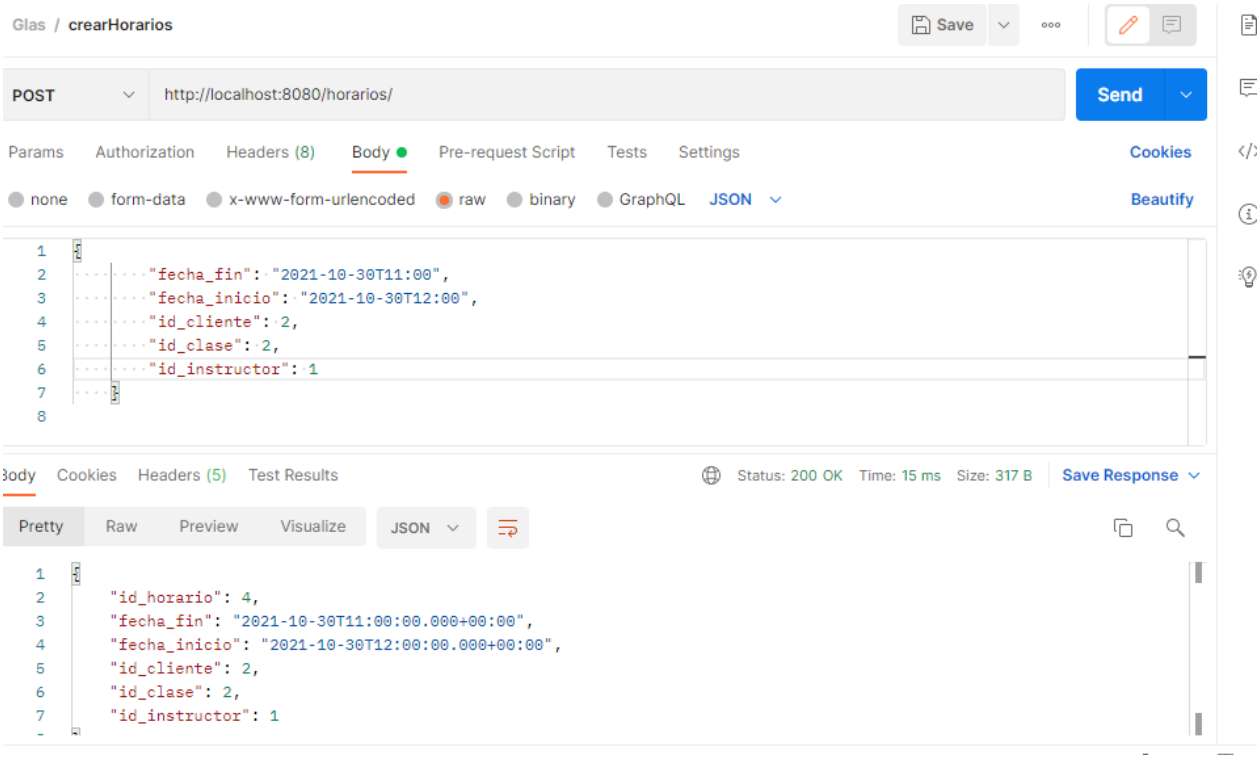
```
2  {
3      "fecha_fin": "2021-10-30T08:00",
4      "fecha_inicio": "2021-10-30T10:00",
5      "id_cliente": 3,
6      "id_clase": 1,
7      "id_instructor": 3
8  }
9
```

Body 200 OK 14 ms 317 B Save Resp

Pretty Raw Preview Visualize JSON

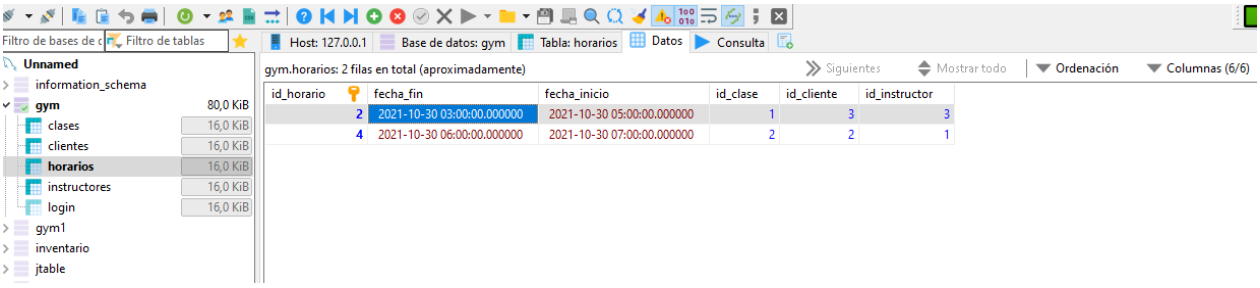
```
2  "id_horario": 2,
3  "fecha_fin": "2021-10-30T08:00:00.000+00:00",
4  "fecha_inicio": "2021-10-30T10:00:00.000+00:00",
5  "id_cliente": 3,
6  "id_clase": 1,
7  "id_instructor": 3
8  }
```

Bootcamp Runner



ConsultarHorarios

Consulta base de datos



Consulta datos postman

GET http://localhost:8080/horarios/ Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (5) Test Results Status: 200 OK Time: 9 ms Size: 473 B Save Response

Pretty Raw Preview Visualize JSON


```
[
    {
        "id_horario": 2,
        "fecha_fin": "2021-10-30T08:00:00.000+00:00",
        "fecha_inicio": "2021-10-30T10:00:00.000+00:00",
        "id_cliente": 3,
        "id_clase": 1,
        "id_instructor": 3
    },
    {
        "id_horario": 4,
        "fecha_fin": "2021-10-30T11:00:00.000+00:00",
        "fecha_inicio": "2021-10-30T12:00:00.000+00:00",
        "id_cliente": 2,
        "id_clase": 2,
```


eliminarHorarios

Se va eliminar el horario con código 4

Glas / New Request Save

DELETE http://localhost:8080/horarios/4 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

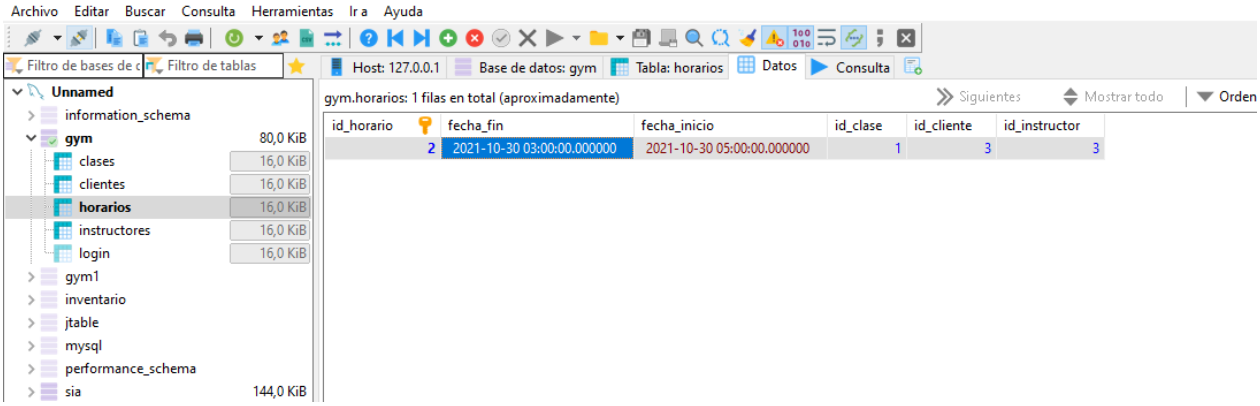
KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (4) Test Results Status: 200 OK Time: 19 ms Size: 123 B Save Response

Pretty Raw Preview Visualize Text

```
1
```

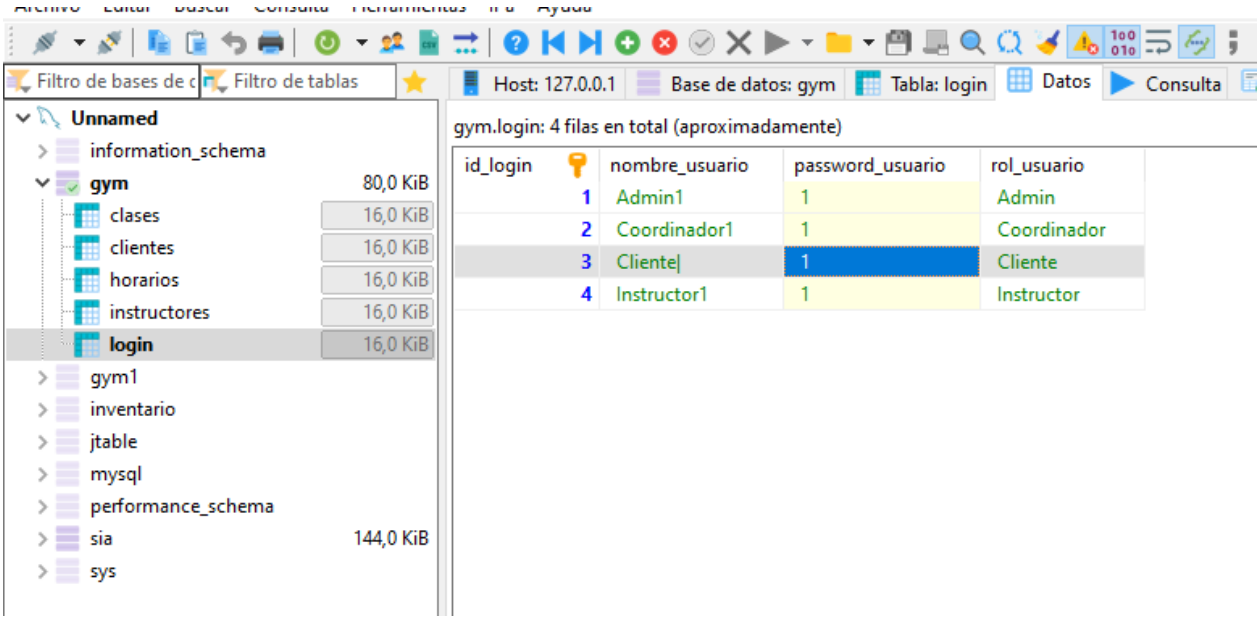
Consulta de base de datos



LOGIN

consultaLogin

Consulta base de datos información.



Glas / login

Save

GET http://localhost:8080/login/ Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (5) Test Results Status: 200 OK Time: 14 ms Size: 535 B Save Response

Pretty Raw Preview Visualize JSON


```
[
    {
        "id_login": 1,
        "nombre_usuario": "Admin1",
        "password_usuario": "1",
        "rol_usuario": "Admin"
    },
    {
        "id_login": 2,
        "nombre_usuario": "Coordinador1",
        "password_usuario": "1",
        "rol_usuario": "Coordinador"
    },
    {
        "id_login": 3,
        "nombre_usuario": "Cliente|",
        "password_usuario": "1",
        "rol_usuario": "Cliente"
    },
```


Glas / login

Save

GET http://localhost:8080/login/ Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (5) Test Results Status: 200 OK Time: 14 ms Size: 535 B Save Response

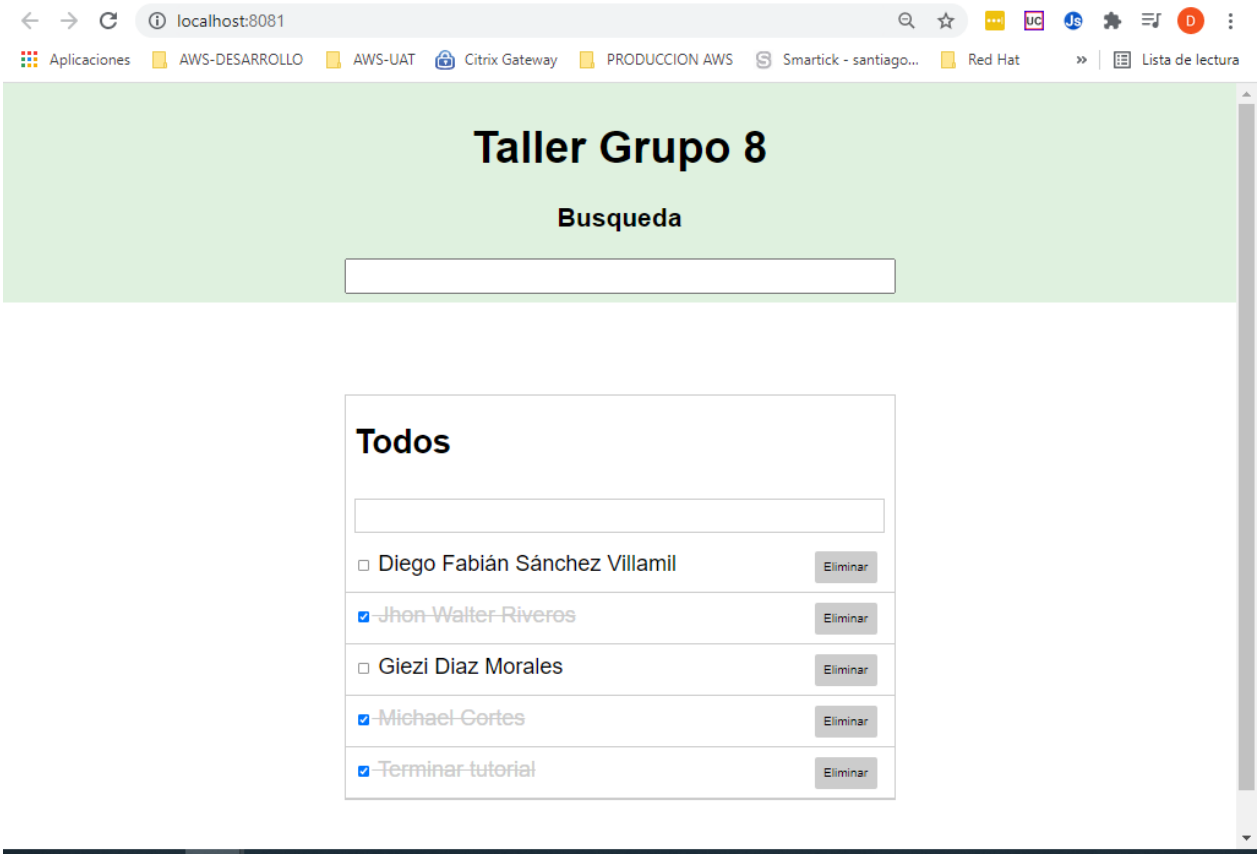
Pretty Raw Preview Visualize JSON


```
    {
        "id_login": 2,
        "nombre_usuario": "Coordinador1",
        "password_usuario": "1",
        "rol_usuario": "Coordinador"
    },
    {
        "id_login": 3,
        "nombre_usuario": "Cliente|",
        "password_usuario": "1",
        "rol_usuario": "Cliente"
    },
    {
        "id_login": 4,
        "nombre_usuario": "Instructor1",
        "password_usuario": "1",
        "rol_usuario": "Instructor"
    }
]
```

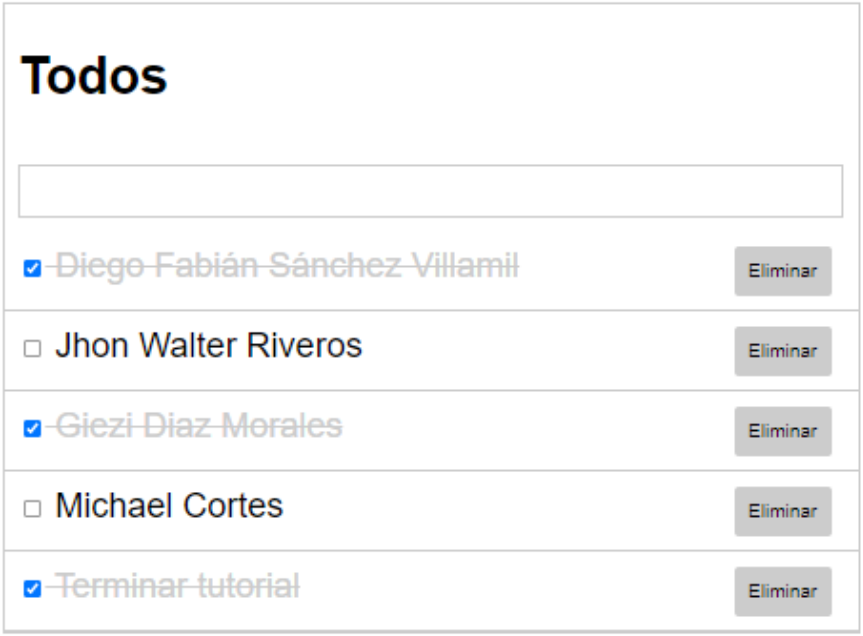
TALLER1

Se desarrolla el taller usando funciones y módulos de vue.

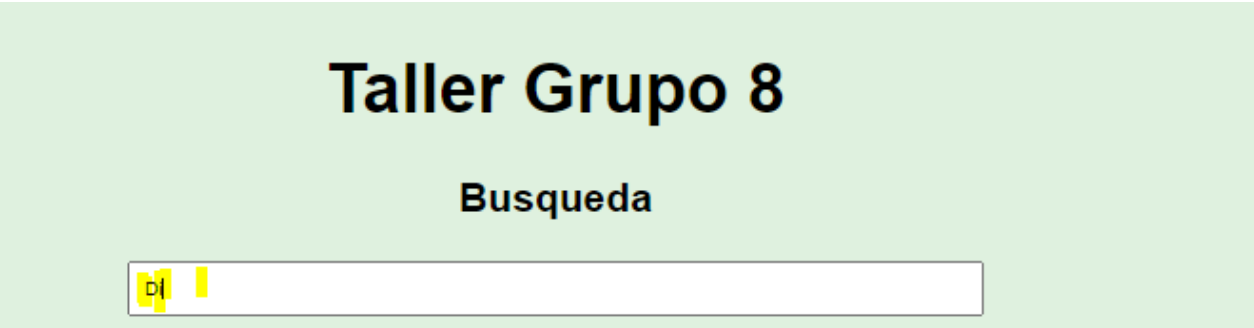
Carga una tabla con un listado de los integrantes del grupo 8. Aparece los que ya cumplieron tareas y los que aun no han cumplido sus tareas.



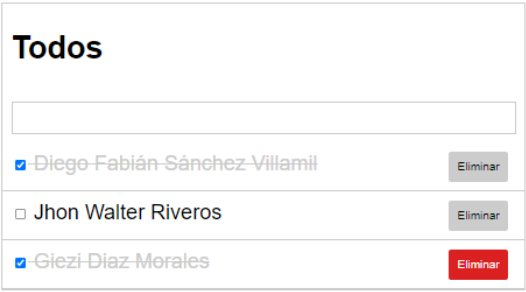
Se puede seleccionar y deseleccionar los integrantes que han cumplido actividades.



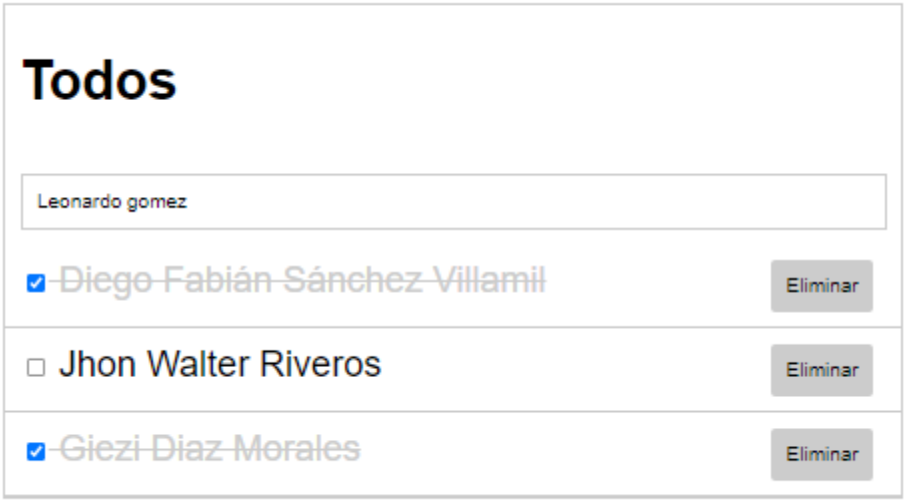
Función que realiza una búsqueda d las coincidencias en el listado de los integrantes del grupo 8

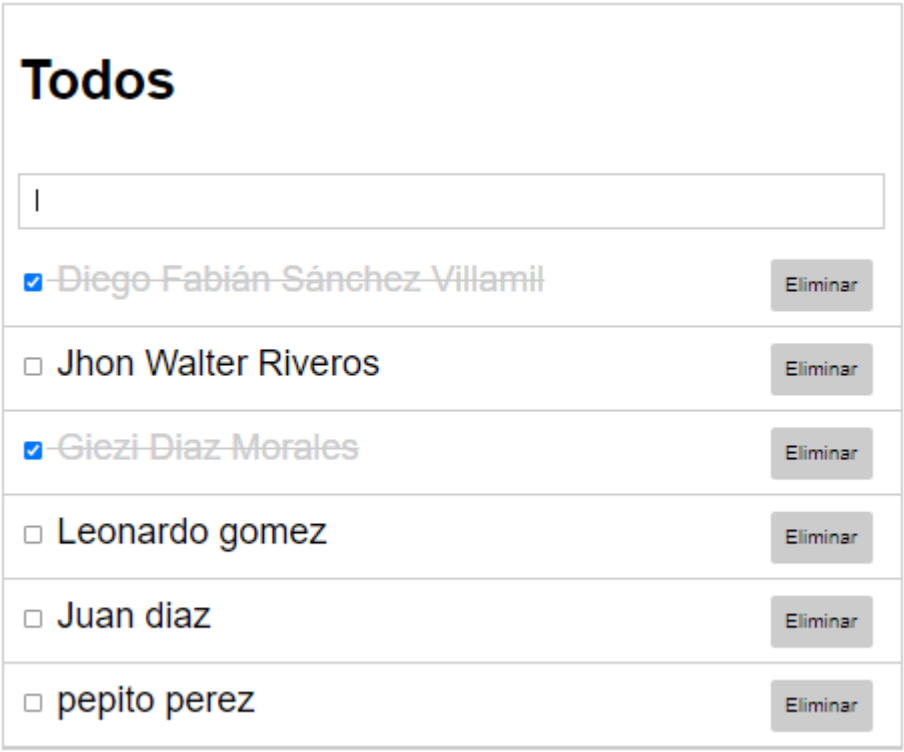


Se puede eliminar los integrantes del grupo



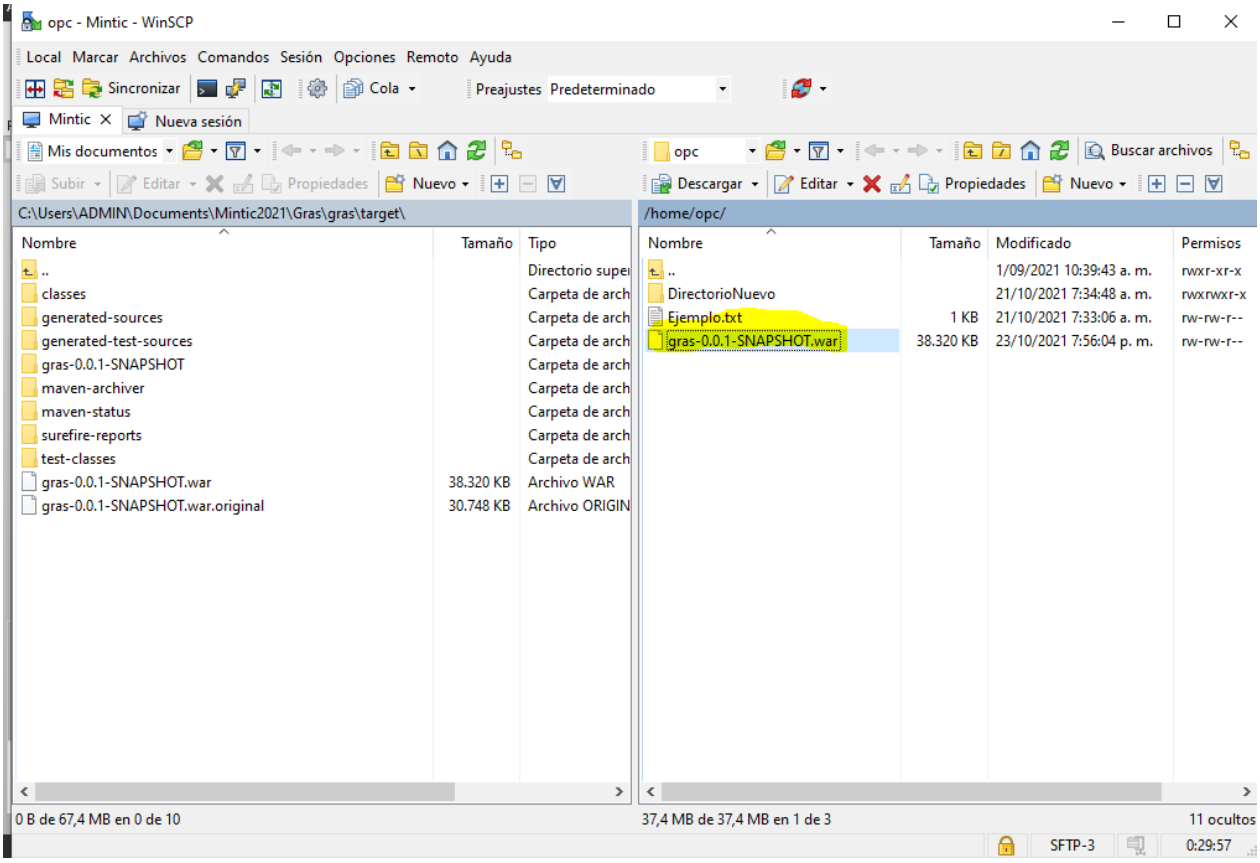
Se pueden agregar más integrantes al grupo.





Despliegue aplicación en el servidor

Nota: Solo se logro desplegar el servicio del backend.



Crear clases usando postman en el servidor dado.

Glas / CrearClases

Save

POST http://132.145.71.76:8080/clases/

Params Auth Headers (8) Body Pre-req. Tests Settings

raw JSON

```
1 {
2     "nombre_clase": "Realizar Aerobics",
3     "descripcion": "Clase aerobics desplegada en el servidor"
4 }
```

Body 200 OK 10 ms 270 B Save Resp

Pretty Raw Preview Visualize JSON

```
1 {
2     "id_clase": 7,
3     "nombre_clase": "Realizar Aerobics",
4     "descripcion": "Clase aerobics desplegada en el servidor"
5 }
```

Consultar la clase creada por el postman en el servidor dado

Glas / **clases**

Save

GET http://132.145.71.76:8080/clases/ Send

Params Auth Headers (6) Body Pre-req. Tests Settings

Body 200 OK 7 ms 272 B Save Resp

Pretty Raw Preview Visualize JSON


```
[
    {
        "id_clase": 8,
        "nombre_clase": "Realizar Aerobics",
        "descripcion": "Clase aerobics desplegada en el servidor"
    }
]
```

SPRINT 6

SPRINT PLANNING

ACT-01
Fecha: 29/10/2021

Horario: 8:15 pm a 8:30 am

Recursos:

Enlace: https://drive.google.com/file/d/1mAuxNzM_JXK9yoFseyxVrsWc3zhGZ3Xg/view?usp=sharing

GitHub: https://github.com/DI3GOS/G8_Proyecto

Trello: https://trello.com/b/lGSvWpi6/g8proyectofinal

Descripción: Asignación actividades Sprint 6 FrontEnd
Participantes:

3. Scrum Master
   - Diego Fabián Sánchez Villamil
4. Development Team
   - Giezi A. Diaz. Morales
   - Jhon Walter Riveros
   - Michael Cortes

DAILY MEETING

ACT-01
Fecha: 30/10/2021

Horario: 8:15 pm a 8:30 am

Recursos:

Enlace: https://drive.google.com/file/d/1KMScP4FI03yBUFZqU3JGP-Y3m2i6D6Dv/view?usp=sharing

GitHub: https://github.com/DI3GOS/G8_Proyecto

Trello: https://trello.com/b/lGSvWpi6/g8proyectofinal

Descripción: revisión de las asignaciones, En donde vimos los diferentes problemas y retrasos de las actividades. Se va a realizar una reunon con el cliente para poder vi
Participantes:

5. Scrum Master
   - Diego Fabián Sánchez Villamil
6. Development Team
   - Giezi A. Diaz. Morales
   - Jhon Walter Riveros
   - Michael Cortes
   -

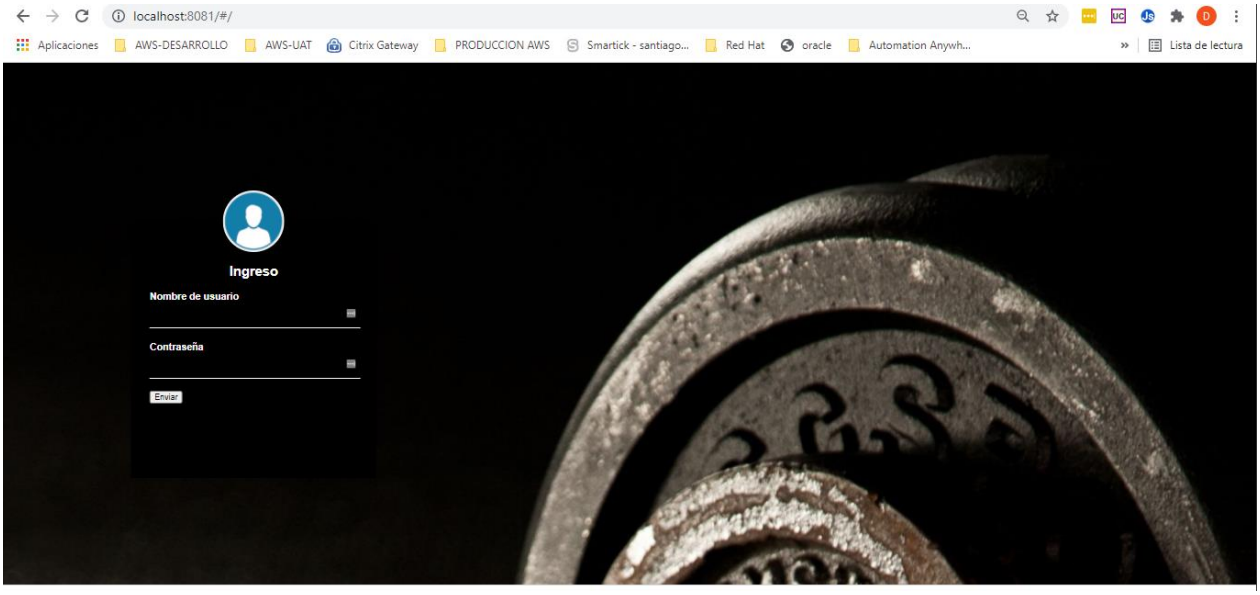
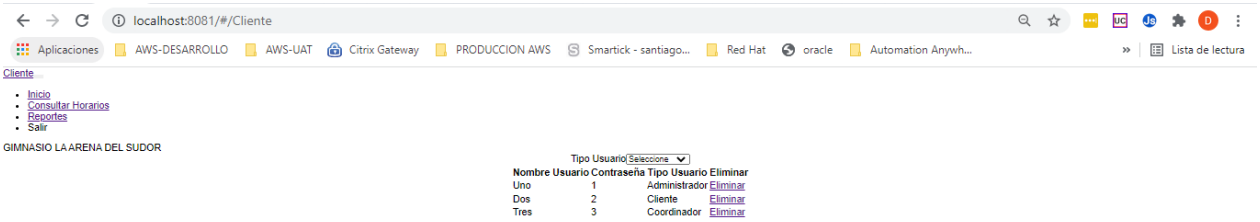
El cuarto ***Sprint*** que tiene una duración de 1 semana, en mi caso lo desarrollaré así: vue.js

	No. Historias de Usuario	*Nombre de la Historia de Usuario*	*Como <tipo de usuario>*	*Quiero <realizar alguna tarea>*	*Para <completar mis tareas a tiempo>*	*Criterios de Aceptación*
	ACT-23	Crear el FrontEnd login	Diego Fabián Sánchez Villamil	Crear el proyecto FrontEnd	Para realizar el proyecto con VUE al lado del Cliente	- Se debe crear el proyecto con todas las dependencias necesarias para la funcionalidad de las peticiones al lado del cliente.
	ACT-24	Crear el Cliente	Giezi A. Diaz. Morales	Crear el proyecto FrontEnd	Para realizar el proyecto con VUE al lado del Cliente	- Se debe crear el proyecto con todas las dependencias necesarias para la funcionalidad de las peticiones al lado del cliente.

Product Increment

CLIENTES

Se realiza el login con la implementación del servicio rest api, post para consultar el login



Realiza el logeo y lo deja según el rol en la pantalla de clientes.