

MANUAL DE USO DEL API DE FUNDACION CARITAS

PROCESO EN LA BASE DE DATOS

HERRAMIENTAS

Lo primero que se debe hacer, es tener instalado dos cosas: Mysql(**Base de Datos**) y MysqlWorkbench (**Gestor de Base de Datos**). Paso siguiente clonar el repositorio de la siguiente URL = https://github.com/unitec-repositorios/Fundacion_Caritas_BE.git (Hacer pull para actualizar)

EMPEZAR

Lo siguiente sera crear el usuario el cual sera el que se usara en la base de datos, por lo cual se llevaran a cabo los siguientes pasos:

1. Crear el usuario que se usara en la base de datos
2. Crear una conexion con el usuario creado
3. Ejecutar el Script en la conexion creada

Nota: Todo lo necesario a hacer a nivel de base de datos esta en la ruta dentro del proyecto /src/SQL/

1. Crear usuario

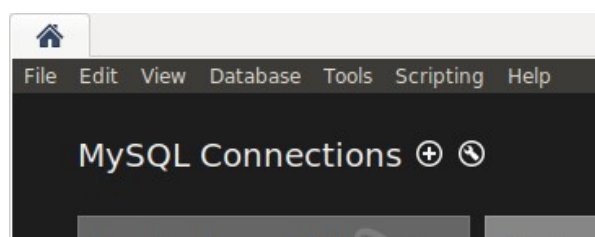
Se necesitara abrir la terminal (ya sea se este en Windows o Linux) y se ejecutara lo siguiente:

En Linux	En Windows
1. sudo mysql	1. Entrar como usuario root en mysql
2. CREATE USER 'caritas'@'localhost' IDENTIFIED BY 'caritas123';	2. CREATE USER 'caritas'@'localhost' IDENTIFIED BY 'caritas123';
3. GRANT ALL PRIVILEGES ON *.* TO 'caritas'@'localhost';	3. GRANT ALL PRIVILEGES ON *.* TO 'caritas'@'localhost';
4. exit	4. exit

2. Crear la conexion

Despues del paso anterior se puede cerrar la consola, lo que viene seria crear la conexion en el gestor de base de datos (**MysqlWorkbench**) y se seguiran los siguientes pasos:

1. Dar Click en el icono de “+”



2. Llenar los datos correspondientes como se muestran; **NOTA:** en el nombre de la conexión se puede poner cualquier nombre; además poner el usuario y contraseña el cual creamos anteriormente:

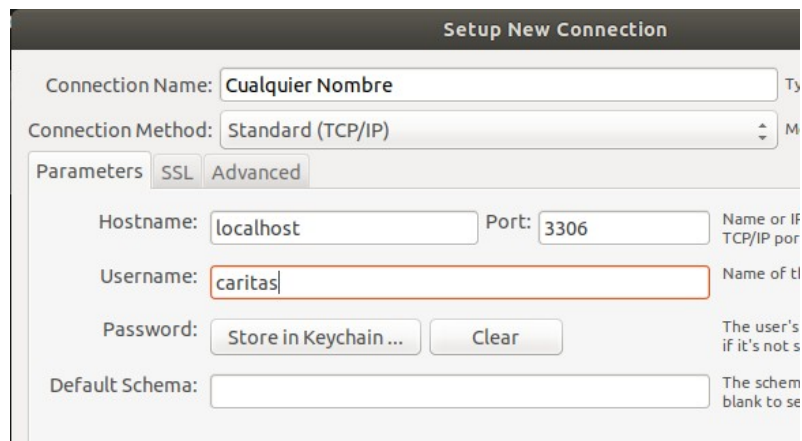
Connection Name: <cualquier nombre>

Hostname: localhost

Port: 3306

Username: caritas

Password: caritas123



3. Ejecutar el Script

Entrar a la conexión creada y luego se abrirá un editor y se verá solamente un esquema que sería **sys**. Luego se va a **File** → **Open SQL Script** y luego seleccionan el archivo que está en la carpeta del proyecto que se llama **SQL** y se llama **BASE_DATOS_CARITAS.sql** y luego ejecutan todo dando click aquí:



Y con eso se finalizaría todo a nivel de base de datos.

API

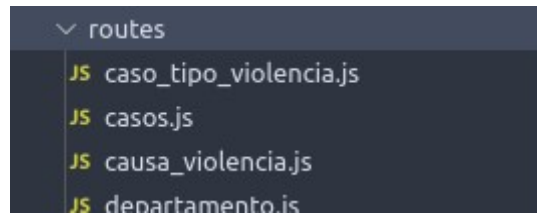
Paquetes

El API solo cuenta con 3 paquetes que se pueden instalar usando **npm install** o instalándolo de manera individual

```
"dependencies": {
  "cors": "^2.8.5",
  "express": "^4.17.1",
  "mysql": "^2.17.1"
```

Rutas y Controladores

En el proyecto hay una carpeta que se llama **Routes** en ella se almacenta diferentes archivos que contiene los controladores de todas las tablas de la base de datos



Enumeremos las rutas y sus controladores:

- **usuarios => <host>/api/usuarios**
- **ubicacion de violencia => <host>/api/ubicacionviolencia**
- **tratamiento => <host>/api/tratamiento**
- **tipo de violencia => <host>/api/tipoviolenia**
- **tipo de condicion => <host>/api/tipocondicion**
- **terapeutas => <host>/api/terapeuta**
- **remision => <host>/api/remision**
- **recursos municipales => <host>/api/recursosmunicipales**
- **pacientes => <host>/api/paciente**
- **municipio => <host>/api/municipio**
- **estado de ocupacion => <host>/api/estadoocupacion**
- **estado civil => <host>/api/estadocivil**
- **estado de atencion => <host>/api/estadoatencion**
- **educacion => <host>/api/educacion**
- **departamento => <host>/api/departamento**
- **causa de violencia => <host>/api/causaviolenia**
- **casos => <host>/api/caso**
- **tipo de violencia por caso => <host>/api/casotipoviolenia**

NOTA: Para estandarizar los metodos de **POST** y **PUT** se usaran request mediante body, eso quiere decir que toda informacion que se crea o se modifica debe tener un cuerpo, En cual en este caso se les manda un JSON, los cuales son bastantes especificos. Por lo cual en la carpeta del API **/src/JSON_SCRIPTS** esta un archivo que se debe agregar al proyecto actual y este contiene funciones que retornan la estructura del JSON adecuado para los **POST** y **PUT request**; entonces cada vez que se ejecute un request que requiera de un

BODY, antes se le mandaran todos los campos como argumentos al la funcion que se requiera en ese momento y lo que retorne la funcion se mandara en el body usando FETCH o Axios

Ejemplo:

1. Incluir el archivo

```
const app = express();
const format = require('./JSON_Scripts/JSON_Formats');
```

2. Usar Culquiera de las funciones que hay en el SCRIPT file y llamarlas para generar el body adecuado para el metodo

```
fetch(url, { // en url iria la url del controlador
  method: 'POST', // Metodo
  body: format.UBICACION_VIOLENCIA_POST_Y_PUT('argumento 1'), // Cualquier funcion disponible en formate
  headers:{
    'Content-Type': 'application/json' // el tipo de que se le mandara
  }
}).then(res => res.json())
.catch(error => console.error('Error:', error))
.then(response => console.log('Success:', response));
```

```
const params = format.USUARIO_POST('argumento1','argumento2');

axios.post('http://localhost/react/submit.php', params, {
  headers: {
    'content-type': 'application/json',
  }
});|
```

NOTA: Ver las funciones que tiene el archivo **JSON_Formats** para ver cual usar dependiendo el request (<nombre del la variable>.<metodo>(argumentos))

NOTA 2: Revisar cuantos parametros se tiene cada funcion(por lo general solo es 1 a excepcion de pacientes y casos) a su misma vez en la estructura de controladores se adjuntara un imagen con los campos que tiene dicha tabla (los que tiene id se ignora ya que es autoincremental entonces no van como parametros de las funciones para tranformar a JSON)

ESTRUCTURA DE CONTROLADORES

USUARIOS

usuario	contraseña
mariorva301	holmundo

GET /api/usuarios

Obtiene todos los usuarios que hay en la base de datos y los nombre con que obtiene son

GET POR ID /api/usuarios/:usuario

Recibe un parametro y retorna el usuario que encontro

POST /api/usuarios

Recibe un BODY que contiene la informacion (usuario y contraseña) para crear el usuario con su respectiva contraseña

PUT /api/usuarios/:USER

Recibe un parametro que es el usuario que se desea modificar y se le manda por BODY la informacion que se desea modificar (Nuevo nombre usuario y la contraseña)

DELETE /api/usuarios/:USER

Recibe de parametro al usuario que se desea eliminar

UBICACION DE VIOLENCIA

Por default hay dos tipos: Rural y Urbana

id_uviolencia	ubicacion
1	Urbana

GET /api/ubicacionviolencia

Obtiene todas las ubicaciones de violencia que hay con su respectivo id

POST /api/ubicacionviolencia

Agrega una nueva ubicacion de violencia y esta recibe un BODY que contiene la ubicacion que se desea agregar

PUT /api/ubicacionviolencia/:ID

Modifica la ubicacion de violencia que es representada por el ID mandada por parametro y recibe un BODY con la nueva informacion a la cual se desea cambiar

DELETE /api/ubicacionviolencia/:ID

Recibe el ID por parametro del elemento que se desea eliminar

TRATAMIENTO

Por default hay un solo tratamiento : Consejeria

id_tratamiento	tratamiento
1	Consejeria

GET /api/tratamiento

Obtiene todos los elementos que son tratamientos con su respectivo id

POST /api/tratamiento

Recibe un BODY que contiene el nombre del tratamiento que se va agregar

PUT /api/tratamiento/:ID

Recibe de parametro el ID del tratamiento y tambien recibe un BODY que contendra la modificacion del tratamiento

DELETE /api/tratamiento/:ID

Recibe de parametro el ID del tratamiento que se desea borrar

TIPO DE VIOLENCIA

Por default los tipos de violencia que hay son: Psicologica, Fisica, Economica y Sexual

id_tipoviolenencia	tipo
1	Violencia Psicologica

GET /api/tipoviolenencia

Obtiene todos los tipos de violencia que hay con su respectivo id

POST /api/tipoviolenencia

Recibe de un BODY que contiene el tipo de violencia que se va a agregar

PUT /api/tipoviolenencia/:ID

Recibe de parametro el ID del tipo de violencia que se desea modificar y tambien recibe un BODY con la modificacion del tipo de violencia

DELETE /api/tipoviolenencia/:ID

Recibe de parametro el ID del tipo de violencia que se desea eliminar

TIPO DE CONDICION

Contiene la condicion del paciente

Por default tiene dos tipos de condicion: Victima y Agresor

id_condicion	condicion
1	Victima
2	Agresor

GET /api/tipocondicion

Obtiene todos los tipo de condicion que hay con su respectivo id

POST /api/tipocondicion

Recibe un BODY el cual contendra el tipo de condicion que se va a agregar

PUT /api/tipocondicion/:ID

Recibe de parametro el ID del tipo de condicion a modificar y recibe un BODY con la modificacion del tipo de condicion

DELETE /api/tipocondicion/:ID

Recibe de parametro el ID del tipo de condicion que se desea eliminar

TERAPEUTAS

Contiene el nombre de los terapeutas los cuales atienden a los pacientes

id_terapeuta	nombre
1	Miriam Fonseca

GET /api/terapeuta

Obtiene todos los terapeutas con su respectivo id

POST /api/terapeuta

Recibe un BODY que contendra un solo dato, que seria el nombre de la terapeuta que se desea agregar

PUT /api/terapeuta/:ID

Recibe un parametro que sera el ID del terapeuta y tambien un BODY que contendra la modificacion del nombre del terapeuta

DELETE /api/terapeuta/:ID

Recibe un parametro que sera el ID del terapeuta que se desea eliminar

REMISION

Contiene la informacion de que tipo de juez remite el caso

Por default estan: JEVD , JEVS, JP

id_remision	juez
1	JEVD

GET /api/remision

Obtiene todos los tipos de jueces con su respectivo id

POST /api/remision

Recibe de BODY el tipo de juez que se desea agregar

PUT /api/remision/:ID

Recibe de parametro un ID que es del tipo de juez al cual se desea modificar y tambien un BODY que contiene la informacion nueva

DELETE /api/remision/:ID

Recibe de parametro un ID del tipo de juez que se desea eliminar

RECURSOS MUNICIPALES

Tambien conocido como el acceso a la justicia que tiene las personas

Por default tiene los siguientes datos: UMEP, OMM, ONG , Juzgado, Fiscalia

id_recursos	tipo
1	UMED

GET /api/recursosmunicipales

Obtiene todos los recursos municipales con sus respectivos id

POST /api/recursosmunicipales

Recibe un BODY el cual tendra la informacion del tipo de recurso municipal que se desea agregar

PUT /api/recursosmunicipales/:ID

Recibe de parametro el ID del recurso que se desea modifica y tambien un BODY con la nueva informacion

DELETE /api/recursosmunicipales/:ID

Recibe de parametro el ID del recurso que se desea eliminar

PACIENTES

id_paciente	identidad	nombres	apellidos	edad	genero	oficio	id_estadoc	id_estado	id_educacion	id_departamento
PATIENTE	PATIENTE	PATIENTE	PATIENTE	PATIENTE	PATIENTE	PATIENTE	PATIENTE	PATIENTE	PATIENTE	PATIENTE

GET /api/paciente

Obtiene toda la lista de pacientes que hay con su respectivo id y tambien cambiando los otros codigos por datos sacados por esos IDs

GET /api/paciente/:ID

Recibe de parametro el ID del paciente y obtiene los codigos de las tablas asociadas con ella (id del estado civil, id remuneracion, id de educacion, id de departamento)

POST /api/paciente

Recibe un BODY que contendra la informacion en el siguiente orden (identidad, nombres, apellido, edad, genero, oficio, id del estado civil, id remuneracion, id de educacion, id de departamento) del nuevo paciente a agregar

PUT /api/paciente/:ID

Recibe un parametro que es el ID del paciente que se desea modificar y tambien recibe un BODY con los datos actualizados (**NOTA: Es el mismo BODY del POST**)

DELETE /api/paciente/:ID

Recibe de parametro el ID del paciente que se desea eliminar

MUNICIPIOS

Se refiere a la localizacion de las parroquias del cual procede el beneficiario

Por default hay: Villanueva, La Lima, San Pedro Sula

id_municipio	nombre
1	San Pedro Sula

GET /api/municipio

Obtiene todos los municipios que estan registrado con su respectivo id

POST /api/municipio

Recibe un BODY el cual contiene el nombre del nuevo municipio a agregar

PUT /api/municipio/:ID

Recibe un parametro ue es el ID del municipio y tambien recibe un BODY que contendra la modifciacion del municipio

DELETE /api/municipio/:ID

Recibe un parametro que es el ID del municipio que se desea eliminar

ESTADO DE OCUPACION

Se refiere a si presenta remuneracion con respecto a su oficio

Por default hay dos: Trabajo remunerado y no remunerado

id_estado	tipo
1	Trabajo Remunerado

GET /api/estadoocupacion

Obtiene todos los estados de ocupacion que hay con su respectivo id

POST /api/estadoocupacion

Recibe un BODY el cual contendra el tipo de remuneracion que se desea agregar

PUT /api/estadoocupacion/:ID

Recibe un parametro que es el ID del tipo de remuneracion que queremos modificar y un BODY con el tipo modificado

DELETE /api/estadoocupacion/:ID

Recibe de parametro el ID del tipo de remuneracion que se quiere eliminar

ESTADO CIVIL

Por default hay: Soltero, Casado y Union libre

id_estadoc	estado
1	Casado

GET /api/estadocivil

Obtiene todos los estados civiles que hay con su respectivo id

POST /api/estadocivil

Recibe un BODY que tiene el estado civil que se quiere ingresar

PUT /api/estadocivil/:ID

Recibe como parametro el ID del estado que se quiere modificar y tambien un BODY que tiene el estado modificado

ESTADO DE ATENCION

Se refiere a si un paciente esta en Proceso, Activo, o Abandono

id_estadoa	estado
1	Proceso

GET /api/estadoatencion

Obtiene todos los estados de atencion que hay con su respectivo id

POST /api/estadoatencion

Recibe un BODY el cul contedra el nuevo estado que se desea agregar

PUT /api/estadoatencion/:ID

Recibe un parametro que es el ID del estado que se quiere modificar y un BODY que contiene el estado modificado

DELETE /api/estadoatencion/:ID

Recibe de parametro el ID del estado que se quiere eliminar

EDUCACION

Hace referencia a los niveles educativos que hay

id_educacion	tipo
1	Analfabeta

GET /api/educacion

Obtiene todos los tipo de educacion que hay

POST /api/educacion

Recibe un BODY que contiene el tipo de nivel educativo que se desea agregar

PUT /api/educacion/:ID

Recibe un parametro que es el ID del tipo de nivel educativo que se desea modificar y tambien un BODY con el elemento modificado

DELETE /api/educacion/:ID

Recibe un parametro que es el ID del tipo de nivel educativo que quiere eliminar

Se refiere a los departamentos del país residente

GET /api/departamento

POST /api/departamento

Recibe un BODY que contiene el nombre del nuevo departamento a agregar

Recibe un parametro que hace referencia al ID del departamento que desea modificar y tambien un BODY que contiene el nuevo nombre del departamento

Recibe un parametro que hacer referencia al ID del departamento que desea eliminar

Hace referencia a que se dio el motivo de la violencia

Por default tiene: Economica, Infidelidad, Alcholismo

GET /api/causaviolencia

Obtiene todas las causas de violencia con su respectivo id

POST /api/causaviolencia

Recibe un BODY que contiene la causa de violencia que desea agregar

PUT /api/causaviolencia/:ID

Recibe un parametro que es el ID de la causa de violencia que se desea modificar y tambien un BODY que tiene el nombre de la nueva causa

DELETE /api/causaviolencia/:ID

Recibe un parametro que es el ID de la causa de violencia que desea eliminar

id_caso	numero_expediente	cantidad_beneficiados	id_uviolencia	id_estadoa	id_municipio	id_remision	id_recursos	id_causa	id_terapeuta	id_paciente	id_condicion	id_tratamient
---------	-------------------	-----------------------	---------------	------------	--------------	-------------	-------------	----------	--------------	-------------	--------------	---------------

GET /api/caso

Obtiene todos los casos con su respectivo id y modificando los id de otras tablas por sus respectivos campos

GET /api/caso/:ID

Requiere un parametro que hace referencia al ID del caso, y lo que hace es obtener los codigos de las tablas externas afiliadas de el caso encontrado por el ID

POST /api/caso

Requiere un BODY que contendra los elementos del caso en su respectivo orden (numero de expediente, cantidad de beneficiados, id ubicacion de violencia, id estado de atencion, id municipio, id remision, id recursos municipales, id causa de violencia, id de la terapeuta, id del paciente, id del tipo de condicion, id del tratamiento) para crear un caso

PUT /api/caso/:ID

Requiere un parametro que hara referencia al ID del caso al cual se quiere modificar y tambien recibe la misma estructura del BODY del POST con los valores nuevos

DELETE /api/caso/:ID

Requiere un parametro que hace referencia al ID del caso que se desea eliminar

TIPOS DE VIOLENCIA POR CASO

Esta tabla surgio debido a que un caso puede tener varios tipos de violencia

id_caso	id_tipoviolen
---------	---------------

GET /api/casotipoviolen/:ID

Recibe un parametro que es el ID del caso, lo que hace que retorne los tipos de violencia que tiene dicho caso

POST /api/casotipoviolen

Recibe un BODY que solamente contiene codigos haciendo referencia al ID del caso y al ID del tipo de violencia que se va agregar al caso

PUT /api/casotipoviolen

Recibe el mismo BODY del POST haciendo referencia siempre el ID del caso que se quiere modificar y el otro ID del tipo de violencia que se quiere cambiar

DELETE /api/casotipoviolen/:IDCASO

Recibe de parametro el ID del caso al cual se quiere eliminar los tipo de violencia