

## ***Link al repositorio:***

**Github:** [amaya7398/sistema-clinico-microservicios \(github.com\)](https://github.com/amaya7398/sistema-clinico-microservicios)

Aquí se podrán encontrar los diagramas de entidad-relación de la DB como también el de clases

## ***Roles y tareas:***

**Developer:** Daniel Amaya

- Desarrollo de features
- Creación de clases e interfaces
- Creación de servicios
- Comunicación con el DB-manager
- Diagramas de los servicios creados

**DB-manager:** Daniel Amaya

- Desarrollo de diagramas de entidad relación de la base de datos
- Scripts para la creación de tablas, inserts, etc
- Administración de la estructura de las DBs
- Comunicación con el developer
- Creación de las bases de datos [clínica-X, clínica-Y, clínica-Z]

**Devops:** Daniel Amaya

- Configuraciones de los diversos servicios
  - Puertos, inputs, outputs,
- Despliegue de los servicios
- Integración de los servicios
- Creación de los DTOS

Por falta de tiempo se desarrolló...

*API: Admin\_Personal (port:8080)*

Hecho:

- Ficha\_identificacion

Faltó:

- Agenda (Se debería poder agendar una cita al cliente con ID de la ficha)

*API: Antecedentes\_cliente (port:8081)*

Hecho:

- antecedentes\_paciente
- antec\_heredo\_fam

## Base de datos MySQL:

### SCRIPTS: Para db de Microservicio de “Admin-personal”.

Creación de la base de datos “clínica-admin-personal”

```
CREATE DATABASE `clinica-admin-personal`
```

Creación de la tabla “ficha\_identificacion”

```
-- `clinica-admin-personal`.ficha_identificacion definition

CREATE TABLE `ficha_identificacion` (
  `num_expediente` bigint NOT NULL,
  `apellidos` varchar(60) NOT NULL,
  `direccion` varchar(60) DEFAULT NULL,
  `email` varchar(30) DEFAULT NULL,
  `escolaridad` varchar(20) DEFAULT NULL,
  `estado_civil` varchar(20) DEFAULT NULL,
  `fecha_creacion_expediente` date NOT NULL,
  `fecha_nacimiento` date NOT NULL,
  `nombre` varchar(40) NOT NULL,
  `ocupacion` varchar(20) DEFAULT NULL,
  `telefono` bigint NOT NULL,
  PRIMARY KEY (`num_expediente`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Obtención de una ficha\_identificacion

```
Hibernate:
select
  f1_0.num_expediente,f1_0.apellidos,f1_0.direccion,f1_0.email,f1_0.escola
  ridad,f1_0.estado_civil,f1_0.fecha_creacion_expediente,f1_0.fecha_nacimi
  ento,f1_0.nombre,f1_0.ocupacion,f1_0.telefono
from ficha_identificacion f1_0
where f1_0.num_expediente=?
```

Actualización de una “ficha\_identificación”

```
Hibernate:
select next_val as id_val from ficha_identificacion_seq for update
```

```
Hibernate:
update ficha_identificacion_seq set next_val= ? where next_val=?
```

Agregar datos a la base de datos

```
Hibernate:
insert into ficha_identificacion
  (apellidos,direccion,email,escolaridad,estado_civil,fecha_creacion_exped
  iente,fecha_nacimiento,nombre,ocupacion,telefono,num_expediente)
values (?,?,?,?,?,?,?,?,?,?,?,?,?)
```

**SCRIPTS: Para db de Microservicio “antecedentes-personal”**

Creación de la base de datos “clínica-antecedentes”

```
CREATE DATABASE `clinica-antecedentes`
```

Creación de la tabla “antecedentes\_paciente”

```
-- `clinica-antecedentes`.antecedentes_paciente definition

CREATE TABLE `antecedentes_paciente` (
  `id` bigint NOT NULL,
  `expediente_id` bigint NOT NULL,
  `fecha_creado` date NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Obtención de antecedentes

```
Hibernate:
Select
    a1_0.id,a1_0.expediente_id,a1_0.fecha_creado
from antecedentes_paciente a1_0
where a1_0.expediente_id=?
```

Agregar datos a la base de datos

```
Hibernate: insert into antecedentes_paciente (expediente_id,fecha_creado,id)
values (?, ?, ?)
```

Actualizar datos de “antecedentes”

```
Hibernate:
Select
    a1_0.id,a2_0.antecedente_id,a2_0.alergias,a2_0.cardiopatias,a2_0.diabetes,a2_0.nefropatias,a2_0.neurologicas,a2_0.otros,a2_0.psiquiatricos,a1_0.expediente_id,a1_0.fecha_creado
from antecedentes_paciente a1_0
left join antec_heredo_fam a2_0 on a1_0.id=a2_0.antecedente_id
where a1_0.id=?

Hibernate:
update antecedentes_paciente set expediente_id=?,fecha_creado=? where id=?
```

## Creación de la base de datos “antec\_heredo\_fam”

```
-- `clinica-antecedentes`.antec_heredo_fam definition

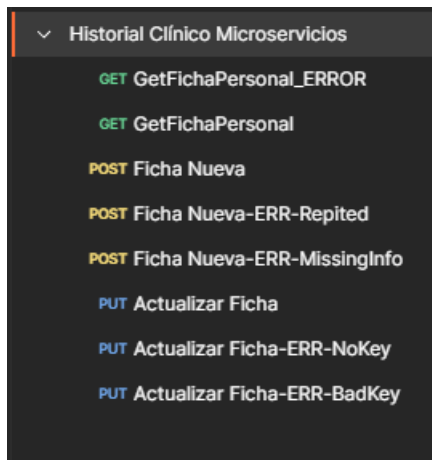
CREATE TABLE `antec_heredo_fam` (
  `antecedente_id` bigint NOT NULL,
  `alergias` varchar(255) DEFAULT NULL,
  `cardiopatias` varchar(255) DEFAULT NULL,
  `diabetes` varchar(255) DEFAULT NULL,
  `nefropatias` varchar(255) DEFAULT NULL,
  `neurologicas` varchar(255) DEFAULT NULL,
  `otros` varchar(255) DEFAULT NULL,
  `psiquiatricos` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`antecedente_id`),
  CONSTRAINT `FKhgpihdsv52ba8okvxobn68egm` FOREIGN KEY (`antecedente_id`)
REFERENCES `antecedentes_paciente` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

## Obtención de “antecedents heredo familiares”

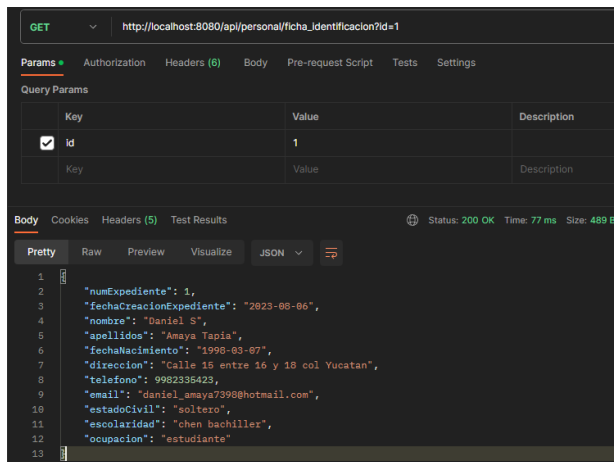
```
Hibernate:
Select
a1_0.id,a2_0.antecedente_id,a2_0.alergias,a2_0.cardiopatias,a2_0.diabete
s,a2_0.nefropatias,a2_0.neurologicas,a2_0.otros,a2_0.psiquiatricos,a1_0.
expediente_id,a1_0.fecha_creado
from antecedentes_paciente a1_0
left join antec_heredo_fam a2_0 on a1_0.id=a2_0.antecedente_id
where a1_0.id=?
```

## Peticiones API desde PostMan:

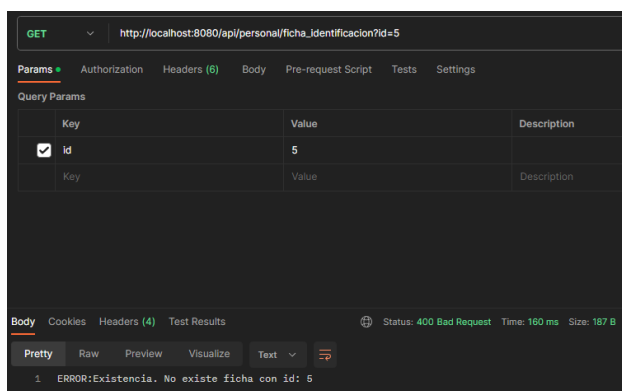
Endpoints: Para API de “Admin-personal”.



CRUD “ficha\_identificacion” del paciente



GET => ficha personal con ID correcta:



GET => ficha personal con ID incorrecta (no existe):

POST http://localhost:8080/api/personal/ficha\_identificacion

Body (JSON):

```

1 {
2   "fechaCreacionExpediente": "2023-08-05",
3   "nombre": "Persona 4",
4   "apellidos": "Persona 4 apellidos",
5   "fechaNacimiento": "1998-03-07",
6   "direccion": "Calle 15 entre 16 y 18 col Yucatan",
7   "telefono": 9982335423,
8   "email": "persona4@hotmail.com",
9   "estadoCivil": "soltero",
10  "escolaridad": "Persona 4 estudio",
11  "ocupacion": "Persona 4 chamba"
12 }

```

Status: 201 Created Time: 42 ms Size: 506 B

Body (Pretty):

```

1 {
2   "numExpediente": 152,
3   "fechaCreacionExpediente": "2023-08-05",
4   "nombre": "Persona 4",
5   "apellidos": "Persona 4 apellidos",
6   "fechaNacimiento": "1998-03-07",
7   "direccion": "Calle 15 entre 16 y 18 col Yucatan",
8   "telefono": 9982335423,
9   "email": "persona4@hotmail.com",
10  "estadoCivil": "soltero",
11  "escolaridad": "Persona 4 estudio",
12  "ocupacion": "Persona 4 chamba"
13 }

```

POST => Creación de ficha personal correctamente:

POST http://localhost:8080/api/personal/ficha\_identificacion

Body (JSON):

```

1 {
2   "fechaCreacionExpediente": "2023-08-05",
3   "nombre": "Persona 3",
4   "apellidos": "Persona 3 apellidos",
5   "fechaNacimiento": "1998-03-07",
6   "direccion": "Calle 15 entre 16 y 18 col Yucatan",
7   "telefono": 9982335423,
8   "email": "persona3@hotmail.com",
9   "estadoCivil": "soltero",
10  "escolaridad": "Persona 3 estudio",
11  "ocupacion": "Persona 3 chamba"
12 }

```

Status: 400 Bad Request Time: 90 ms Size: 200 B

Body (Text):

```

1 ERROR:Existencia | Ya existe una ficha de identificacion

```

POST => ERROR: Ficha existente. No se puede repetir

POST http://localhost:8080/api/personal/ficha\_identificacion

Body (JSON):

```

1 {
2   "fechaCreacionExpediente": "2023-08-05",
3   "apellidos": "Persona 3 apellidos",
4   "direccion": "Calle 15 entre 16 y 18 col Yucatan",
5   "telefono": 9982335423,
6   "email": "persona3@hotmail.com",
7   "estadoCivil": "soltero",
8   "escolaridad": "Persona 3 estudio",
9   "ocupacion": "Persona 3 chamba"
10 }

```

Status: 400 Bad Request Time: 154 ms Size: 267 B

Body (Text):

```

1 not-null property references a null or transient value : edu.uady.administrativo.model.FichaId
  fechaNacimiento

```

POST => ERROR: Falta información necesaria:

PUT ▼ http://localhost:8080/api/personal/ficha\_identificacion

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL ● JSON ▼

```

1  {
2    "numExpediente": 1,
3    "fechaCreacionExpediente": "2023-08-06",
4    "nombre": "Daniel S",
5    "apellidos": "Amaya Tapia",
6    "fechaNacimiento": "1998-03-07",
7    "direccion": "Calle 15 cruzamiento 16 y 18",
8    "telefono": 9982335423,
9    "email": "daniel_amaya7398@hotmail.com",
10   "estadoCivil": "soltero",
11   "escolaridad": "chen bachiller",
12   "ocupacion": "estudiante"
13 }

```

body Cookies Headers (5) Test Results ⊕ Status: 201 Created Time: 336 ms Size: 488 B

Pretty Raw Preview Visualize JSON ▼

```

1  {
2    "numExpediente": 1,
3    "fechaCreacionExpediente": "2023-08-06",
4    "nombre": "Daniel S",
5    "apellidos": "Amaya Tapia",
6    "fechaNacimiento": "1998-03-07",
7    "direccion": "Calle 15 cruzamiento 16 y 18",
8    "telefono": 9982335423,
9    "email": "daniel_amaya7398@hotmail.com",
10   "estadoCivil": "soltero",
11   "escolaridad": "chen bachiller",
12   "ocupacion": "estudiante"
13 }

```

PUT => Actualizar correcta (id no null, id existe, toda la info se añade)

PUT ▼ http://localhost:8080/api/personal/ficha\_identificacion

Params Authorization Headers (0) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL ● JSON ▼

```

1  {
2    "fechaCreacionExpediente": "2023-08-06",
3    "nombre": "Daniel S",
4    "apellidos": "Amaya Tapia",
5    "fechaNacimiento": "1998-03-07",
6    "direccion": "Calle 15 entre 16 y 18 col Yucatan",
7    "telefono": 9982335423,
8    "email": "daniel_amaya7398@hotmail.com",
9    "estadoCivil": "soltero",
10   "escolaridad": "chen bachiller",
11   "ocupacion": "estudiante"
12 }

```

body Cookies Headers (4) Test Results ⊕ Status: 400 Bad Request Time: 8 ms Size: 190 B

Pretty Raw Preview Visualize Text ▼

```

1  ERROR:Petición | No se proporciono ID de ficha

```

PUT => ERROR: ID insertado == null

PUT ▼ http://localhost:8080/api/personal/ficha\_identificacion

Params Authorization Headers (0) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL ● JSON ▼

```

1  {
2    "numExpediente": 10,
3    "fechaCreacionExpediente": "2023-08-06",
4    "nombre": "Daniel S",
5    "apellidos": "Amaya Tapia",
6    "fechaNacimiento": "1998-03-07",
7    "direccion": "Calle 15 entre 16 y 18 col Yucatan",
8    "telefono": 9982335423,
9    "email": "daniel_amaya7398@hotmail.com",
10   "estadoCivil": "soltero",
11   "escolaridad": "chen bachiller",
12   "ocupacion": "estudiante"
13 }

```

body Cookies Headers (4) Test Results ⊕ Status: 400 Bad Request Time: 11 ms Size: 189 B

Pretty Raw Preview Visualize Text ▼

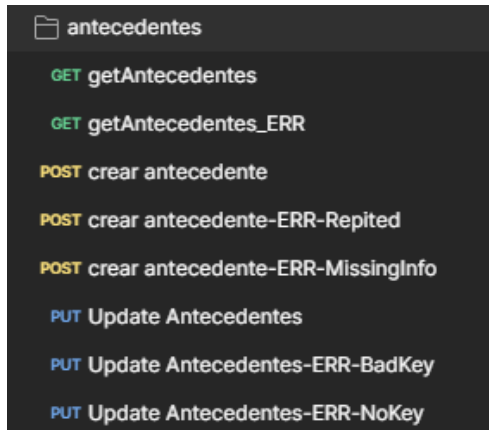
```

1  ERROR:Existencia | No existe ficha con id: 10

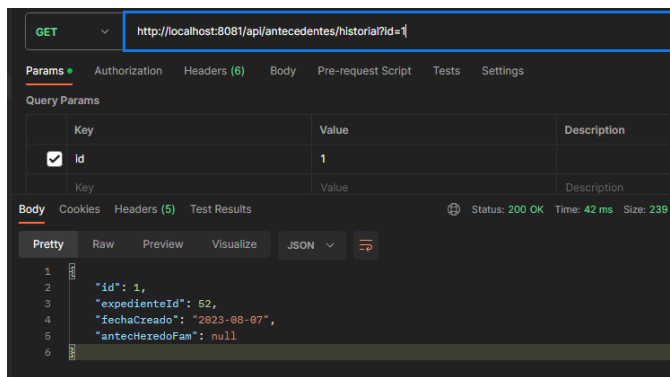
```

PUT => ERROR: ID insertado no existe en el sistema

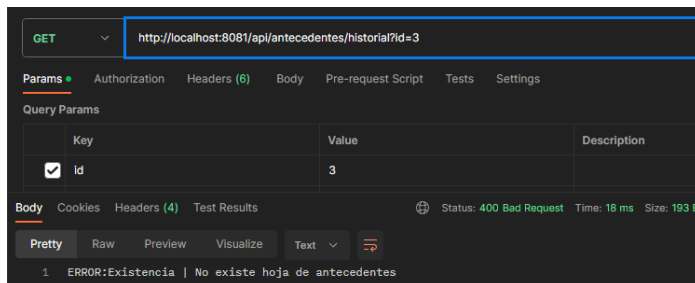
## Endpoints: Para API de “antecedentes”.



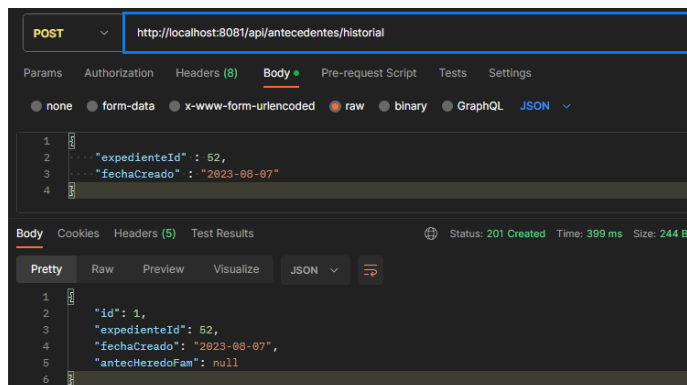
CRUD “antecedentes” del paciente



GET => antecedentes personales con ID correcta:

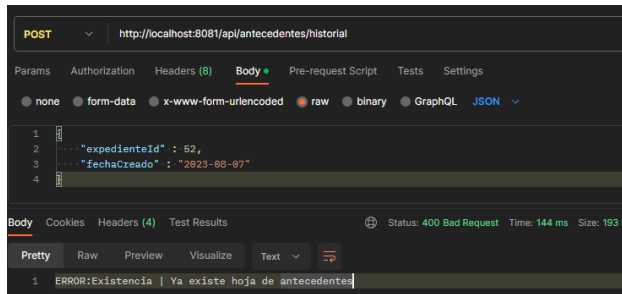


GET => antecedentes personales con ID incorrecta (no existe):

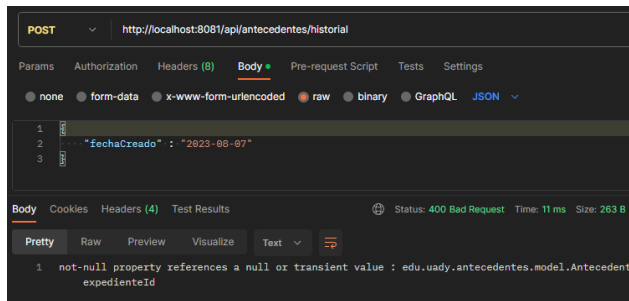


POST => Creación de antecedente personal correctamente:

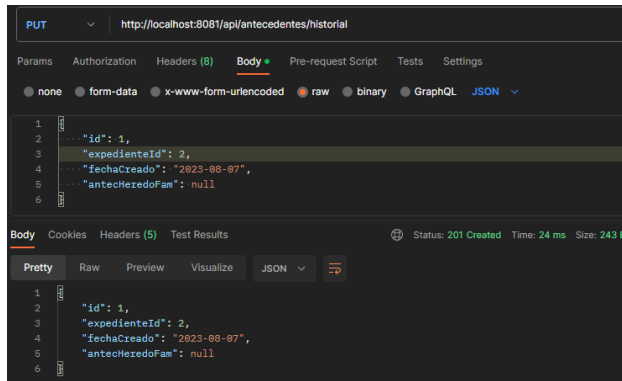




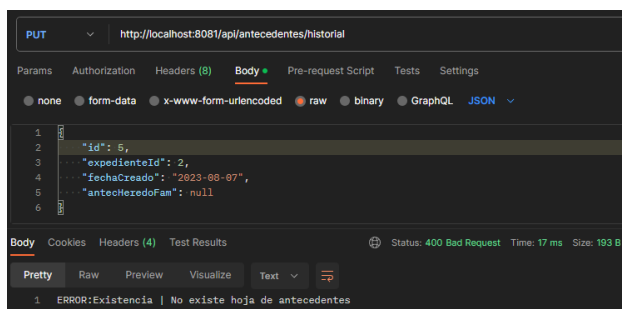
**POST => ERROR:** Antecedente existente. No se puede repetir



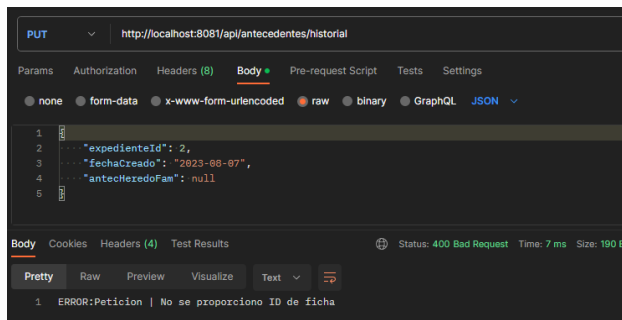
**POST => ERROR:** Falta información necesaria



**PUT => Actualización correcta** (id no null, id existe, toda la info se añade)



**PUT => ERROR:** ID insertado no existe en el sistema



**PUT => ERROR:** ID == null