



Arquitectura de software - Grupo 5

## Integrantes

- RODRIGUEZ BETTY
- DOMINGUEZ OSCAR
- VILLAMARIN VICTOR
- GUAMILAMA NICOLAS
- POAQUIZA MARCO
- TENEMAZA ALANIS

## 💻 Microservicio de Gestión de Usuarios (Citas Médicas)

---

Este proyecto es un microservicio desarrollado en **Laravel** que implementa una **API RESTful** para gestionar usuarios (pacientes) dentro de un sistema de Citas Médicas. Utiliza el patrón **MVC** y **Laravel Sanctum** para la autenticación basada en tokens.

## 🌐 Flujo de Trabajo

El desarrollo se organiza en fases asignadas a estudiantes:

1. **Nicolás:** Configuración inicial del proyecto y la DB.
  2. **Betty:** Estructura de la Base de Datos y Seguridad (Sanctum).
  3. **Víctor:** Definición de Rutas API (Endpoints).
  4. **Valeria:** Lógica del Controlador (Implementación del CRUD).
  5. **Adrián:** Lógica de Validación (Form Requests).
  6. **Marco:** Autenticación (Login/Tokens) y Pruebas Finales.
- 

## 🚀 1. Inicialización del Proyecto (Nicolás)

Estos comandos crean la base del proyecto Laravel y configuran el entorno de trabajo.

1. **Crear el Proyecto Laravel:**

```
composer create-project laravel/laravel microservicio-citas-medicas
```

## 2. Acceder al Directorio:

```
cd microservicio-citas-medicas
```

## 3. Configurar la Base de Datos:

Abrir y editar el archivo `.env` con las credenciales de su base de datos (Ejemplo):

```
# Variables de conexión a la Base de Datos
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=pry_users_microservicio
DB_USERNAME=root
DB_PASSWORD=secret
```

---

## Microservicio de Gestión de Citas y Historial Médico

[USER\_G5\_AS]

Este proyecto es un microservicio Backend desarrollado en Laravel (PHP) con una arquitectura RESTful API. Su objetivo principal es gestionar la autenticación de usuarios (pacientes y doctores), la programación de citas médicas y el registro del historial clínico y tratamientos.

### 🔗 Arquitectura y Diseño

El microservicio opera bajo un enfoque de API Versioning y Protección de Rutas.

- Tecnología Principal: Laravel (con PHP 8+).
- Autenticación: Laravel Sanctum (Token-based authentication) para asegurar todas las rutas sensibles.
- Versionamiento: Todas las rutas se agrupan bajo el prefijo `/api/v1/`.
- Roles: El sistema diferencia entre dos tipos de usuarios:
- Pacientes: Pueden registrarse, iniciar sesión y crear/listar sus propias citas.
- Doctores: Además de lo anterior, pueden crear, actualizar y eliminar entradas de historial médico y tratamientos.

### Estructura de Endpoints (v1)

Funcionalidad	Prefijo/Ruta	Autenticación
Identidad	/v1/login , /v1/logout	Público / Protegido
Citas	/v1/citas	Protegido (Auth:Sanctum)
Historial	/v1/historial	Protegido (Solo Doctores)

## ⚙️ Configuración y Puesta en Marcha

Sigue estos pasos para levantar el microservicio en tu entorno local.

1. Instalación de Dependencias

### 1. Clona el repositorio

```
git clone <URL_DE_TU_REPOSITORIO> USER_G5_AS cd USER_G5_AS
```

### 2. Instala las dependencias de PHP

```
composer install
```

### 3. Copia el archivo de configuración .env

```
cp .env.example .env
```

### 4. Genera la clave de aplicación

```
php artisan key:generate
```

## 2. Configuración de Base de Datos

Asegúrese de configurar las variables de conexión a tu base de datos MySQL en el archivo .env.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=microservicio_usuarios
DB_USERNAME=root
DB_PASSWORD=
```

localhost:8012/phpmyadmin/index.php?route=/sql&pos=0&db=sistemacitasmedicas2&table=tratamientos

phpMyAdmin

Servidor: localhost > Base de datos: sistemacitasmedicas2 > Tabla: tratamientos

Examinar (actual) Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Seguimiento

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0028 segundos.)

SELECT \* FROM `tratamientos`

Perfilando [ Editar en línea ] [ Editar ] [ Explicar SQL ] [ Crear código PHP ] [ Actualizar ]

id	paciente_id	historial_medico_id	doctor_id	nombre_tratamiento	descripcion	fecha_inicio	fecha_fin	estado	created_at	updated_at
----	-------------	---------------------	-----------	--------------------	-------------	--------------	-----------	--------	------------	------------

Operaciones sobre los resultados de la consulta

Crear vista

Guardar esta consulta en favoritos

Etiqueta:  Permitir que todo usuario pueda acceder a este favorito

Guardar esta consulta en favoritos

## Migraciones y Seeders (Población de Datos)

Ejecuta las migraciones para crear las tablas y luego ejecuta los seeders para poblar los datos de prueba esenciales (usuarios, doctores, historial).

### Ejecuta las migraciones

```
php artisan migrate
```

```
php artisan make:migration create_appointments_table
php artisan make:migration create_doctors_table
php artisan make:migration create_patients_table
php artisan make:migration create_schedules_table
php artisan make:migration create_appointment_statuses_table
```

Mostrando filas 0 - 3 (total de 4. La consulta tardó 0.0007 segundos.)

SELECT \* FROM `users`

	id	name	email	email_verified_at	password	fecha_nacimiento	sexo
<input type="checkbox"/>	1	Ana López	ana@ejemplo.com	NULL	\$2y\$12\$VBU6YqPoi2rsLu4bkQbOcxpyQi5.GpdzT/lW8g1y...	1995-03-20	Fer
<input type="checkbox"/>	2	Paciente.JC	paciente.JC@citas.com	NULL	\$2y\$12\$1R.X141FvjobfifaGrJ0eFkhFiHvGoXjgAwMQVmTBx...	1995-03-20	Fer
<input type="checkbox"/>	3	Juan Pérez	juan@ejemplo.com	NULL	\$2y\$12\$KQfgIiDXQlwax9pUHD7e8QycWU3mMAvCRTF/4lFA3...	1985-05-15	Ma
<input type="checkbox"/>	4	Nicolas Guamalama	enguamalama@espe.edu.ec	NULL	\$2y\$12\$/4bLo9VS31mYrwicsl5vDOyfpgewM5O2lbyYQMid8Lz...	1990-01-01	Ma

## Ejecuta los seeders para poblar la base de datos

Incluye: UsuarioSeeder, DoctorSeeder, HistoriaMedicaSeeder

```
php artisan db:seed
```

```
php artisan make:seeder AppointmentSeeder
php artisan make:seeder DoctorSeeder
php artisan make:seeder PatientSeeder
php artisan make:seeder AppointmentStatusSeeder
```

## 4. Ejecución del Servidor

Inicia el servidor local de Laravel.

```
php artisan serve
```

```
Nicolas@Nico MINGW64 ~/Documents/29692-ARQUITECTURA DE SOFTWARE/USER_G5_AS (main)
$ php artisan serve

INFO Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server

2025-11-08 18:08:36 /api/v1/login ..... ~ 511.95
2025-11-08 18:12:14 /api/v1/citas ..... ~ 0.19
2025-11-08 18:22:11 /api/v1/historial/1 ..... ~
2025-11-08 18:23:36 /api/v1/login ..... ~ 0.27
2025-11-08 18:24:12 /api/v1/historial/1 ..... ~ 0.40
```

El microservicio estará accesible en: <http://127.0.0.1:8000>

### 🔗 Datos de Acceso de Prueba (Seeders)

Utiliza las siguientes credenciales para probar la API con Postman o Insomnia. Todos los passwords son password.

HTTP <http://localhost:8000/api/v1/users>

GET <http://localhost:8000/api/v1/users>

Params Authorization **Headers (9)** Body Scripts Settings Cookies

Headers 7 hidden

Key	Value	Description	Bulk Edit	Presets
<input checked="" type="checkbox"/> Accept	application/json			
<input checked="" type="checkbox"/> Authorization	Bearer 3 w3QDf6LipgKa9Walc7NP6YFr...			

Body Cookies Headers (7) Test Results +  
200 OK • 280 ms • 1.59 KB • [Raw](#) [JSON](#) [Preview](#) [Visualize](#)

```
18 |     "id": 2,
19 |     "name": "Paciente.JC",
20 |     "email": "paciente.JC@citas.com",
21 |     "email_verified_at": null,
22 |     "fecha_nacimiento": "1995-03-20T00:00:00.000000Z",
23 |     "sexo": "Femenino",
24 |     "numero_seguro": "AXA98765",
25 |     "historial_medico": "Ninguno conocido.",
26 |     "contacto_emergencia": "555-123-0000",
27 |     "created_at": "2025-11-08T16:27:00.000000Z",
28 |     "updated_at": "2025-11-08T16:27:00.000000Z"
29 },
30 {
31     "id": 3,
32     "name": "Juan Pérez",
33     "email": "juan@ejemplo.com",
34     "email_verified_at": null,
35     "fecha_nacimiento": "1985-05-15T00:00:00.000000Z",
36     "sexo": "Masculino",
37     "numero_seguro": "MED123456",
38     "historial_medico": "Diabetes tipo 2",
39     "contacto_emergencia": "987654321",
40     "created_at": "2025-11-08T16:29:51.000000Z",
41     "updated_at": "2025-11-08T16:29:51.000000Z"
42 },
43 {
44     "id": 4,
45     "name": "Nicolas Guamialama",
46     "email": "enguamialama@espe.edu.ec",
47     "email_verified_at": null,
48     "fecha_nacimiento": "1990-01-01T00:00:00.000000Z",
49     "sexo": "Masculino",
50     "numero_seguro": "SEG123456",
51     "historial_medico": "Sin historial",
52     "contacto_emergencia": "123456789",
53     "created_at": "2025-11-08T16:39:37.000000Z",
54     "updated_at": "2025-11-08T16:39:37.000000Z"
```

Postbot Runner Start Proxy Cookies Vault Trash

The screenshot shows the Postman interface with a successful API call to `http://localhost:8000/api/v1/login`.

**Request:**

- Method: POST
- URL: `http://localhost:8000/api/v1/login`
- Body (raw JSON):

```
1 {
2     "email": "enguamialama@espe.edu.ec",
3     "password": "pasword2551"
4 }
```

**Response:**

- Status: 200 OK
- Time: 375 ms
- Size: 729 B
- Content:

```
1 {
2     "status": "success",
3     "message": "Inicio de sesión exitoso.",
4     "access_token": "1|9i4HQ04SUtsV0hJrYSnFftmPERpHZJpaPvaYoHE2fcceb34d0",
5     "token_type": "Bearer",
6     "user": {
7         "id": 4,
8         "name": "Nicolas Guamialama",
9         "email": "enguamialama@espe.edu.ec",
10        "email_verified_at": null,
11        "fecha_nacimiento": "1990-01-01T00:00:00.000000Z",
12        "sexo": "Masculino",
13        "numero_seguro": "SEG123456",
14        "historial_medico": "Sin historial",
15        "contacto_emergencia": "123456789",
16        "created_at": "2025-11-08T16:39:37.000000Z",
17        "updated_at": "2025-11-08T16:39:37.000000Z"
18    }
19 }
```

doctora@test.com

Necesario para crear/ver Historial

paciente@test.com

Necesario para crear Citas

🔗 Endpoints Clave para Pruebas (Postman)

## A. Autenticación (Público)

URL `http://127.0.0.1:8000/api/v1/`

## Método POST

### Descripción REGISTRO

/api/v1/login

Obtener el token (e.g., usando doctora@test.com).

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** http://127.0.0.1:8000/api/v1/login
- Body:** JSON (showing the request body)
- Response:** 200 OK (435 ms, 706 B)
  - Status:** success
  - Message:** Inicio de sesión exitoso.
  - access\_token:** "1|j6621tP3wNhNgql2zjvYCEh8zWmSH4obgYEloSnC8500df91"
  - token\_type:** Bearer
  - User Data:** (Detailed below)

**Request Body (JSON):**

```
1 {  
2   "email": "doctora@test.com",  
3   "password": "password"  
4 }
```

**Response Body (JSON):**

```
1 {  
2   "status": "success",  
3   "message": "Inicio de sesión exitoso.",  
4   "access_token": "1|j6621tP3wNhNgql2zjvYCEh8zWmSH4obgYEloSnC8500df91",  
5   "token_type": "Bearer",  
6   "user": {  
7     "id": 2,  
8     "name": "Doctora Ana Garcia",  
9     "email": "doctora@test.com",  
10    "email_verified_at": null,  
11    "fecha_nacimiento": "1985-11-20T00:00:00.000000Z",  
12    "sexo": "Femenino",  
13    "numero_seguro": null,  
14    "historial_medico": null,  
15    "contacto_emergencia": "555-5678",  
16    "created_at": "2025-11-09T00:59:43.000000Z",  
17    "updated_at": "2025-11-09T00:59:43.000000Z"  
18  }  
19 }
```

### B. Gestión de Historial (Solo Doctor)

REQUISITO: Usar el Token de doctora@test.com en el Authorization: Bearer Header.

URL http://127.0.0.1:8000/api/v1/historial/1

## Método GET

## Descripción LOGIN

/api/v1/historial/1

POST

Crea un nuevo registro de historia. Requiere paciente\_id, sintomas, diagnostico.

The screenshot shows the Postman interface with the following details:

- Request URL:** http://127.0.0.1:8000/api/v1/historial/1
- Method:** POST
- Headers:** Authorization: Bearer 2|8qCdwf6CuPHTl5R0o52k... (selected)
- Body:** JSON response (200 OK)

```
[{"id": 1, "paciente_id": 1, "doctor_id": 1, "fecha_registro": "2025-10-20T00:00:00.000000Z", "sintomas": "Fiebre persistente durante 3 días, tos seca y dolor de garganta. No hay dificultad respiratoria.", "diagnostico": "Faringitis viral, se recomienda reposo y tratamiento sintomático.", "created_at": "2025-11-09T00:59:43.000000Z", "updated_at": "2025-11-09T00:59:43.000000Z", "doctor": {"id": 1, "user_id": 2, "numero_licencia": "LIC-123456", "especialidad": "Cardiología", "dias_disponibles": "[{"Lunes": "08:00-12:00"}, {"Miércoles": "14:00-18:00"}]"}, "created_at": "2025-11-09T00:59:43.000000Z", "updated_at": "2025-11-09T00:59:43.000000Z"}]
```

## Contribuciones y Soporte

Para contribuciones o si encuentras algún problema de arquitectura, por favor revisa los siguientes archivos clave:

Rutas: routes/api.php

## Controladores:

. app/Http/Controllers/HistoriaMedicaController.php y app/Http/. .Controllers/CitaController.php

## Seeders:

. database/seeders/HistoriaMedicaSeeder.php

The screenshot shows the Postman interface with a collection named "My Workspace". A specific API endpoint, "http://localhost:8000/api/v1/users", is selected. The "Headers" tab is active, showing two headers: "Accept" set to "application/json" and "Authorization" set to "Bearer 3|w3QDf6LipgKa9Walc7NP6YFr...". The "Body" tab shows a JSON response with the following data:

```

{
    "id": 2,
    "name": "Paciente.JC",
    "email": "paciente.J@citas.com",
    "email_verified_at": null,
    "fecha_nacimiento": "1995-03-20T00:00:00.000000Z",
    "sexo": "Femenino",
    "numero_seguro": "AXA98765",
    "historial_medico": "Ninguno conocido."
}
  
```

## 2. Estructura y Dependencias (Betty)

**Nota:** Betty debe editar el archivo de migración preexistente

(`database/migrations/*_create_users_table.php`) para añadir los campos específicos de Citas Médicas antes de migrar.

### 1. Ejecutar Migraciones:

```
php artisan migrate
```

### 2. Instalar Laravel Sanctum:

```
composer require laravel/sanctum
php artisan vendor:publish --
provider="Laravel\Sanctum\SanctumServiceProvider"
```

## 3. Creación de Componentes (Víctor, Valeria, Adrián)

Estos comandos generan los archivos principales del patrón MVC (Controladores) y de Validación (Form Requests).

### 1. Crear Controladores API (Víctor):

```
php artisan make:controller UserController --api  
php artisan make:controller AuthController
```

### 2. Crear Form Requests (Adrián):

```
php artisan make:request StoreUserRequest  
php artisan make:request UpdateUserRequest
```

---

## 4. Puesta en Marcha y Pruebas (Marco)

Una vez que todos los archivos anteriores contengan el código de implementación, ejecute el servidor.

### 1. Iniciar el Servidor de Desarrollo:

```
php artisan serve
```

### 2. Endpoints Clave para Postman (Marco):

- **Login (Público):** POST <http://127.0.0.1:8000/api/login> (Retorna el **Bearer Token**).
- **Crear Usuario (Protegido):** POST <http://127.0.0.1:8000/api/usuarios>
- **Listar Usuarios (Protegido):** GET <http://127.0.0.1:8000/api/usuarios>
- **Actualizar Usuario (Protegido):** PUT/PATCH <http://127.0.0.1:8000/api/usuarios/{id}>

**Importante:** Todas las rutas del CRUD ([/api/usuarios](#)) requieren el encabezado **Authorization: Bearer <TOKEN>** para funcionar.