

## Contrato de Comunicación entre Backend y Frontend

### Endpoint 1: Obtener Usuarios

- **Nombre del Endpoint:** /usuarios
- **Tipo de Solicitud:** GET
- **Parámetros:**
  - Ninguno
- **Contenido Esperado:**

```
[  
  {  
    "id": 1,  
    "nombre": "Juan",  
    "apellido": "Pérez",  
    "email": "juan@example.com"  
  },  
  {  
    "id": 2,  
    "nombre": "Maria",  
    "apellido": "González",  
    "email": "maria@example.com"  
  }  
]
```

**Rol de Autorización:** veterinario, admin

**Descripción:** Devuelve una lista de usuarios, cada uno con su nombre, apellido y email.

## Endpoint 2: Obtener Pacientes

- **Nombre del Endpoint:** /pacientes
- **Tipo de Solicitud:** GET
- **Parámetros:**
  - Ninguno
- **Contenido Esperado:**

```
[  
  {  
    "id": "1",  
    "nombre": "Firulais",  
    "fechaNacimiento": "2020-01-15",  
    "especie": "Perro",  
    "raza": "Labrador",  
    "sexo": "Macho",  
    "estado": "Atendido",  
    "dueñold": "3"  
  },  
  {  
    "id": "2",  
    "nombre": "Michi",  
    "fechaNacimiento": "2018-05-10",  
    "especie": "Gato",  
    "raza": "Persa",  
    "sexo": "Hembra",  
    "estado": "En atención",  
    "dueñold": "5"
```

```
}  
]
```

**Rol de Autorización:** usuario, veterinario, admin

**Descripción:** Devuelve una lista de citas con información del cliente, paciente, fecha y hora.

#### Endpoint 4: Eliminar Cita

- **Nombre del Endpoint:** /citas/{id}
- **Tipo de Solicitud:** DELETE
- **Parámetros:**
  - id: (número entero) ID de la cita que se desea eliminar.
- **Contenido Esperado:**
  - Respuesta vacía o mensaje de éxito.

```
{  
  
  "message": "Cita eliminada con éxito"  
  
}
```

**Rol de Autorización:** veterinario

**Descripción:** Elimina una cita específica identificada por su id. Se requiere confirmación antes de proceder con la eliminación.

### Endpoint 5: Obtener Pacientes por Usuario

- **Nombre del Endpoint:** /pacientes
- **Tipo de Solicitud:** GET
- **Parámetros:**
  - Ninguno
- **Contenido Esperado:**

```
[  
  {  
    "id": 1,  
    "nombre": "Firulais",  
    "especie": "Perro",  
    "clienteid": 1  
  },  
  {  
    "id": 2,  
    "nombre": "Michi",  
    "especie": "Gato",  
    "clienteid": 1  
  }  
]
```

**Rol de Autorización:** usuario

**Descripción:** Devuelve una lista de los pacientes pertenecientes al usuario logueado. El cliente puede seleccionar entre sus mascotas para agendar una reserva.

#### Endpoint 6: Registrar Reserva

- **Nombre del Endpoint:** /citas
- **Tipo de Solicitud:** POST

#### Contenido Esperado en el Body:

clienteid: (número entero) ID del cliente que está realizando la reserva.

pacienteid: (número entero) ID del paciente (mascota) asociado con la reserva.

fecha: (string) Fecha de la cita en formato YYYY-MM-DD.

hora: (string) Hora de la cita en formato HH:MM.

```
{  
  "clienteid": 1,  
  "pacienteid": 1,  
  "fecha": "2024-10-23",  
  "hora": "08:00"  
}
```

**Rol de Autorización:** usuario

**Descripción:** Permite que el usuario logueado registre una nueva reserva para uno de sus pacientes en una fecha y hora seleccionada.

### Endpoint 7: Obtener Usuario Logueado

- **Nombre del Endpoint:** /usuarios/{id}
- **Tipo de Solicitud:** GET
- **Parámetros:**

id: (número entero) ID del usuario.

- **Contenido Esperado:**

```
{  
  "id": 1,  
  "nombre": "Juan",  
  "apellido": "Pérez",  
  "email": "juan@example.com"  
}
```

**Rol de Autorización:** usuario

**Descripción:** Devuelve la información del usuario logueado para mostrarla en el sistema de reservas. Esta información se utiliza para asociar la reserva con el usuario.

### Endpoint 8: Actualizar Paciente

- **Nombre del Endpoint:** /pacientes/{id}
- **Tipo de Solicitud:** PUT

- **Parámetros:**

id: (string) ID del paciente que se desea actualizar.

**Body:**

```
{  
  "id": "1",  
  "nombre": "Firulais",  
  "fechaNacimiento": "2020-01-15",  
  "especie": "Perro",  
  "raza": "Labrador",  
  "sexo": "Macho",  
  "estado": "En recuperación",  
  "dueñold": "3"  
}
```

**Contenido Esperado en el Body:**

- id: (string) ID del paciente.
- nombre: (string) Nombre del paciente.
- fechaNacimiento: (string) Fecha de nacimiento del paciente en formato YYYY-MM-DD.
- especie: (string) Especie del paciente.
- raza: (string) Raza del paciente.
- sexo: (string) Sexo del paciente.
- estado: (string) Estado actual del paciente (por ejemplo, "En atención", "Recuperación", "De alta").
- dueñold: (string) ID del dueño del paciente.

**Rol de Autorización:** Admin, veterinario , paciente

**Descripción:** Actualiza los datos del paciente identificado por su id en la base de datos.

#### **Endpoint 9: Actualizar Usuario**

- **Nombre del Endpoint:** /usuarios/{id}
- **Tipo de Solicitud:** PUT
- **Parámetros:**

id: (string) ID del usuario que se desea actualizar.

#### **Body:**

```
{  
  "id": "1",  
  "nombre": "Juan",  
  "apellido": "Pérez",  
  "run": "21859421-2",  
  "email": "juan.perez@example.com",  
  "telefono": "987654321"  
}
```

#### **Contenido Esperado en el Body:**

- id: (string) ID del usuario.
- nombre: (string) Nombre del usuario.
- apellido: (string) Apellido del usuario.
- run: (string) RUN del usuario.



- email: (string) Correo electrónico del usuario.
- telefono: (string) Número de teléfono del usuario.

**Rol de Autorización:** admin

**Descripción:** Actualiza la información de un usuario identificado por su id.

#### **Endpoint 10: Crear Nuevo Usuario (Veterinario)**

- **Nombre del Endpoint:** /usuarios
- **Tipo de Solicitud:** POST
- **Parámetros:**

Body:

```
{  
  "id": 3,  
  "nombre": "Carlos",  
  "apellido": "Rodríguez",  
  "direccion": "Avenida Siempre Viva 123",  
  "telefono": "912345678",  
  "email": "carlos.rodriguez@example.com",  
  "run": "21345678-9",  
  "contrasena": "password123",  
  "rol": 2  
}
```

**Contenido Esperado en el Body:**

- id: (número entero) ID generado para el nuevo veterinario.
- nombre: (string) Nombre del veterinario.
- apellido: (string) Apellido del veterinario.
- direccion: (string) Dirección del veterinario.
- telefono: (string) Número de teléfono del veterinario (validación: 9 dígitos).
- email: (string) Correo electrónico del veterinario.
- run: (string) RUN del veterinario (validación: formato XXXXXXXX-X).
- contrasena: (string) Contraseña del veterinario.
- rol: (número entero) Rol del usuario, 2 para veterinario.

**Rol de Autorización:** admin

**Descripción:** Registra un nuevo veterinario con los datos proporcionados.

**Validaciones**

- **RUN:** Se valida que el RUN tenga el formato correcto: XXXXXXXX-X, sin puntos y con guion.

- **Teléfono:** Se valida que el número de teléfono tenga 9 dígitos y comience con un 9 (por ejemplo: 912345678).
- **Contraseña:** Se valida que la contraseña coincida con la confirmación de la contraseña.

### Endpoint 11: Autenticación de Usuarios

- **Nombre del Endpoint:** /usuarios
- **Tipo de Solicitud:** GET
- **Parámetros (Query Params):**
  - email: (string) Correo electrónico del usuario que intenta iniciar sesión.
  - contraseña: (string) Contraseña del usuario.

### Contenido Esperado:

```
[  
  {  
    "id": 1,  
    "nombre": "Juan",
```

```
"apellido": "Pérez",  
"email": "juan.perez@example.com",  
"contrasena": "password123",  
"rol": 2  
}  
]
```

**Rol de Autorización:** usuario, vet, admin

**Descripción:** Verifica las credenciales del usuario enviadas a través de la solicitud. Si las credenciales son correctas, devuelve el usuario y su rol (1: Cliente, 2: Veterinario, 3: Admin). Este rol es usado para redirigir al usuario a la vista correspondiente (/InicioCliente, /InicioVet, /InicioAdmin).

#### **Validaciones**

- **Email:** Se valida que el correo electrónico proporcionado exista en la base de datos.
- **Contraseña:** Se valida que la contraseña proporcionada coincida con la almacenada en la base de datos para ese correo electrónico.
- **Redirección por Rol:**
  - Si el rol es 1 (Cliente), el usuario es redirigido a /InicioCliente.
  - Si el rol es 2 (Veterinario), el usuario es redirigido a /InicioVet.
  - Si el rol es 3 (Admin), el usuario es redirigido a /InicioAdmin.

#### **Endpoint 2: Registrar Nuevo Usuario**

- **Nombre del Endpoint:** /usuarios
- **Tipo de Solicitud:** POST
- **Parámetros:**

**Body:**

```
{  
  "id": 2,  
  "nombre": "Carlos",  
  "apellido": "Rodríguez",  
  "telefono": "912345678",  
  "email": "carlos.rodriguez@example.com",  
  "direccion": "Avenida Siempre Viva 123",  
  "run": "21345678-9",  
  "contrasena": "password123",  
  "rol": 1  
}
```

#### **Contenido Esperado en el Body:**

- id: (número entero) ID generado basado en el conteo de usuarios.
- nombre: (string) Nombre del usuario.
- apellido: (string) Apellido del usuario.
- telefono: (string) Número de teléfono del usuario (validación: 9 dígitos).
- email: (string) Correo electrónico del usuario.
- direccion: (string) Dirección del usuario.
- run: (string) RUN del usuario (validación: formato XXXXXXXX-X).
- contrasena: (string) Contraseña del usuario.
- rol: (número entero) Rol del usuario. Se usa 1 para Cliente en este caso.

**Rol de Autorización:** usuario

**Descripción:** Registra un nuevo usuario en la base de datos con los datos proporcionados.

#### **Validaciones**

- **RUN:** Se valida que el RUN tenga el formato correcto: XXXXXXXX-X, sin puntos y con guion.

- **Teléfono:** Se valida que el número de teléfono tenga 9 dígitos (formato: 9XXXXXXXXX).
- **Contraseña:** Se valida que la contraseña coincida con la confirmación de la contraseña.
- **Correo Electrónico:** Se verifica que el correo no esté ya registrado en la base de datos antes de permitir el registro.