

# Manual Técnico del Sistema de Agendamiento de Citas - "Tu Amigo Peludo"

---

## 1. Información General del Proyecto

- **Nombre del Proyecto:** Sistema de Agendamiento de Citas Veterinarias
  - **Ubicación:** Puente Aranda, Bogotá DC
  - **Tecnologías Utilizadas:**
    - Frontend: Angular 15
    - Backend: .NET 9 (Web API)
    - Base de Datos: SQL Server
- 

## 2. Arquitectura General

El sistema sigue una arquitectura basada en el patrón Cliente-Servidor:

- **Angular:** Se encarga del manejo de formularios, validaciones y consumo de servicios REST.
  - **.NET Web API:** Expone servicios para consultar, registrar y cancelar citas.
  - **SQL Server:** Almacena los datos de clientes, citas, veterinarios y horarios disponibles.
- 

## 3. Estructura del Proyecto

### 3.1 Frontend (Angular)

Estructura de carpetas:

/src

/app

/components

- agendamiento.component.ts
- agendamiento.component.html

/services

- cita.service.ts

app.module.ts

### 3.2 Backend (.NET)

Estructura de carpetas:

/WebTuAmigoPeludo

/Controllers

- AgendarController.cs

/Models

- Cita.cs

/Services

- CitaService.cs

WebTuAmigoPeludo.csproj

---

## 4. Líneas de Código Relevantes

### 4.1 Frontend - `agendamiento.component.ts`

```
consultar() {  
  const { tipoDocumento, documento } = this.formCita.value;  
  this.citaService.consultarCitas(tipoDocumento, documento)  
    .subscribe({  
      next: (resp) => {  
        this.citas = resp;  
        this.mostrarDetalleCita = true;  
      },  
      error: () => {  
        this.mensajePopup = 'No se encontraron citas para ese documento.';  
        this.mostrarPopup = true;  
      }  
    });  
}  
  
cerrarPopup(): void {  
  this.mostrarPopup = false;  
}
```

#### 4.2 Backend - AgendarController.cs

```
[HttpGet("Consultar")]
public IActionResult Consultar(string tipoDocumento, string documento)
{
    var citas = _citaService.ConsultarPorDocumento(tipoDocumento, documento);
    if (!citas.Any())
    {
        return NotFound("No se encontraron citas para ese documento.");
    }
    return Ok(citas);
}
```

#### 4.3 Backend - CitaService.cs

```
public List<Cita> ConsultarPorDocumento(string tipoDocumento, string documento)
{
    return _context.Citas
        .Where(c => c.TipoDocumento == tipoDocumento && c.Documento == documento && c.Estado == 1)
        .ToList();
}
```

---

### 5. Dependencias Clave

#### Angular:

- @angular/forms
- primeng
- rxjs

#### .NET:

- Entity Framework Core
- Microsoft.AspNetCore.Mvc
- Swashbuckle (Swagger)

---

## 6. Ejecución del Proyecto

### Angular:

npm install

ng serve -o

### .NET:

dotnet restore

dotnet build

dotnet run

---

## 7. Recomendaciones Finales

- Hacer backup frecuente de la base de datos.
  - Realizar validaciones del lado del servidor.
  - Mantener la separación de responsabilidades en capas (Controller, Service, Repository).
- 

**Fin del Manual Técnico**