**Proyecto Final**

**Curso:**

**Programación VI**

**Docente:**

**Braulio Sandi**

**Nombre de estudiantes:**

**Brian Morales Obando**

**Steven Fonseca Araya**

**María Arrieta Diaz**

**Fecha de entrega: 14 de agosto de 2025**

**SIREC - Sistema Inteligente de Reservas para la CCSS**

**Grupo: 3**

**Institución:**

**Universidad Metropolitana Castro Carazo**

Tabla de contenido

[Documentación del Proyecto SIREC 2](#_Toc30245)

[Objetivo 2](#_Toc22641)

[Tecnologías Utilizadas 2](#_Toc11734)

*[Frontend](#_Toc32765)* [2](#_Toc32765)

*[Backend](#_Toc31985)* [2](#_Toc31985)

*[Base de datos](#_Toc11405)* [3](#_Toc11405)

*[ORM](#_Toc26691)* [3](#_Toc26691)

*[Reportes](#_Toc23471)* [3](#_Toc23471)

*[Control de versiones](#_Toc15158)* [3](#_Toc15158)

*[Autenticación](#_Toc9970)* [3](#_Toc9970)

[Funcionalidades Implementadas 3](#_Toc23993)

*[Autenticación y Autorización](#_Toc16546)* [3](#_Toc16546)

*[Gestión de Usuarios (Administrador)](#_Toc25245)* [3](#_Toc25245)

*[Asignación de Médicos a Consultorios](#_Toc15156)* [4](#_Toc15156)

*[Gestión de Citas Médicas](#_Toc22983)* [4](#_Toc22983)

*[Reportes](#_Toc6456)* [4](#_Toc6456)

[Arquitectura del Sistema 5](#_Toc3288)

[Modelo de Datos 5](#_Toc20947)

*[Entidades principales](#_Toc16960)* [5](#_Toc16960)

*[Diagrama de Entidades](#_Toc5644)* [6](#_Toc5644)

[Implementación de API REST con ASP.NET Core 6](#_Toc16864)

*[Controladores y Endpoints](#_Toc17773)* [6](#_Toc17773)

*[Autenticación JWT](#_Toc28570)* [7](#_Toc28570)

*[Validaciones de Datos](#_Toc13514)* [7](#_Toc13514)

*[Buenas Prácticas](#_Toc13185)* [7](#_Toc13185)

[Implementación de Angular 17 8](#_Toc16433)

*[Bootstrap 5](#_Toc25748)* [8](#_Toc25748)

[Instrucciones de Instalación y Ejecución 8](#_Toc10565)

[Link GitHub 9](#_Toc21017)

[Referencias 9](#_Toc3479)

Tabla de diagramas

[Diagrama 1 . Entidades 6](#_Toc23747)

# **Documentación del Proyecto SIREC**

El Sistema de Registro de Citas Médicas (SIREC) es una aplicación web completa desarrollada como proyecto final del curso de Programación Avanzada, que integra un frontend en Angular y un backend en ASP.NET Core con Entity Framework y SQL Server. El sistema implementa un flujo completo de gestión de usuarios, roles, citas médicas y consultorios, con autenticación segura mediante JWT y reportes con Crystal Reports.

## **Objetivo**

Diseñar e implementar una solución web para la gestión de citas médicas, permitiendo la administración de usuarios con roles diferenciados (Administrador, Médico, Paciente), asignación de médicos a consultorios, solicitud y gestión de citas, y control de estado de las mismas.

## **Tecnologías Utilizadas**

### ***Frontend***

Angular 17 con Bootstrap 5.

### ***Backend***

ASP.NET Core 6 con REST APIs.

### ***Base de datos***

SQL Server.

### ***ORM***

Entity Framework Core.

### ***Reportes***

Crystal Reports.

### ***Control de versiones***

GitHub.

### ***Autenticación***

JWT (JSON Web Tokens).

## **Funcionalidades Implementadas**

### ***Autenticación y Autorización***

Registro de usuarios con rol por defecto (Paciente) y opción de asignar otros roles desde el panel de administración.

Inicio de sesión con generación de token JWT que incluye ID de usuario y rol.

Cierre de sesión y verificación de autenticación desde el frontend.

Redirección dinámica según rol del usuario (Administrador → /admin, Médico → /medico, Paciente → /dashboard).

### ***Gestión de Usuarios (Administrador)***

CRUD completo para usuarios.

Asignación y edición de roles.

Interfaz con tabla interactiva, iconos Feather y modales Bootstrap.

Validación de campos obligatorios.

### ***Asignación de Médicos a Consultorios***

Listado de médicos disponibles.

Selección de consultorio y médico para asignación.

API para registrar y listar asignaciones.

Validación de datos en backend.

### ***Gestión de Citas Médicas***

#### Paciente. Solicitud de cita seleccionando día, hora y consultorio.

#### Validaciones Backend. Evitar duplicidad en mismo horario y consultorio, verificar horario laboral, asegurar que exista médico asignado.

#### Médico: Vista de citas asignadas y opción de cambiar estado (Pendiente, Realizada, Cancelada) sin eliminar.

#### Administrador. Gestión global de citas.

### ***Reportes***

Reportes generados con Crystal Reports.

Exportación a PDF.

Filtros para rango de fechas y estados de cita.

## **Arquitectura del Sistema**

La arquitectura está basada en el patrón cliente-servidor. El frontend en Angular consume las APIs expuestas por el backend ASP.NET Core. Entity Framework Core se encarga del acceso a datos en SQL Server. Crystal Reports se integra en el backend para la generación de reportes.

## **Modelo de Datos**

### ***Entidades principales***

Usuario (IDUsuario, Nombre, Apellido, Correo, Contraseña, etc.)

Rol (IDRol, Nombre)

UsuarioRol (IDUsuario, IDRol)

Consultorio (IDConsultorio, Nombre, Especialidad, Ubicacion, HorarioAtencion)

Cita (IDCita, Fecha, Hora, Estado, Observaciones, IDUsuario, IDConsultorio)

MedicoConsultorio (ID, IDUsuario, IDConsultorio)

### ***Diagrama de Entidades***

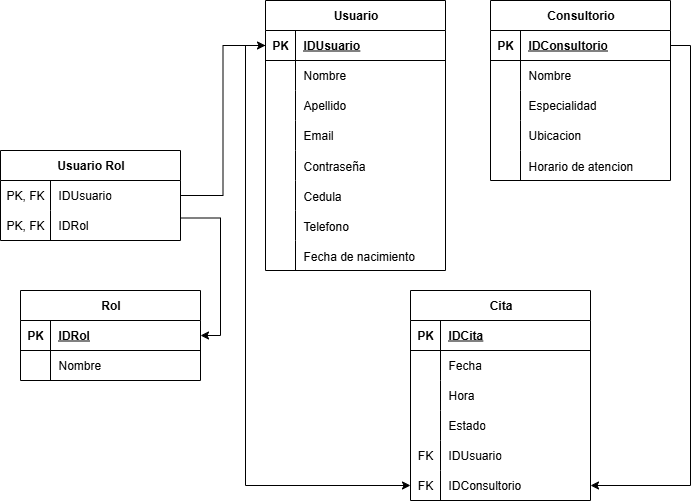


Diagrama 1. Entidades

## **Implementación de API REST con ASP.NET Core**

En esta fase se desarrollarán los controladores y endpoints principales del sistema SIREC, usando ASP.NET Core Web API. Se implementará autenticación con JWT, validaciones con atributos y rutas RESTful para usuarios, citas, consultorios y autenticación.

### ***Controladores y Endpoints***

Dentro de la carpte/Contrtollers se crearon lo siguientes controladores:

UsuarioController → /api/usuarios.

CitaController → /api/citas.

ConsultorioController → /api/consultorios.

AuthController → /api/auth (registro e inicio de sesión)

MedicoConsultorioController →/api/MedicoConsultorio.

### ***Autenticación JWT***

Para proteger rutas y autenticar usuarios se uso JWT (JSON Web Tokens):

Instalar el paquete: Microsoft.AspNetCore.Authentication.JwtBearer.

Configurar en Program.cs la autenticación con JWT.

Proteger controladores o rutas con [Authorize] según el rol del usuario.

Incluir el token en las solicitudes desde Angular mediante un interceptor.

### ***Validaciones de Datos***

Para asegurar la integridad de datos, se debe de usar DataAnnotations en los modelos:

[Required]: Campo obligatorio.

[EmailAddress]: Formato de correo válido.

[StringLength]: Longitud mínima o máxima.

[Range]: Rango numérico válido.

Estas validaciones son detectadas automáticamente por ASP.NET cuando el modelo se recibe en el controlador.

### ***Buenas Prácticas***

Usar rutas con nombre explícito como: (`[Route("api/[controller]")]`).

Utilizar DTOs para entrada y salida de datos si se necesita separar del modelo base.

Documentar las rutas usando Swagger (ya viene configurado en nuevas plantillas).

Proteger las rutas de escritura (POST/PUT/DELETE) con autorización.

Validar errores con ModelState.IsValid y retornar códigos HTTP apropiados.

## **Implementación de Angular 17**

Esta etapa con Angular contiene la parte visual del proyecto, desde el login y el registro de usuarios, donde el paciente puede gestionar sus citas, el administrador posee el acceso completo y todo lo que conlleva de consultorios, usuarios, citas y acceso a reportes, y los médicos pueden ver sus citas asignadas y marcarlas como atendidas, esto con implementación del framework Bootstrap para mejorar la vista del usuario final.

### ***Bootstrap 5***

Su función es la de ayudar a mejorar visualmente el proyecto con ayuda de los componentes que este ofrece.

Su instalación se realiza con los siguientes pasos.

Por medio de la terminal ejecutando npm install bootstrap.

Dentro del archivo angular.Json en la sección Styles se agrega la siguiente íinea de código node\_modules/bootstrap/dist/css/bootstrap.min.css

## **Instrucciones de Instalación y Ejecución**

1. Clonar el repositorio desde GitHub.
2. Configurar la cadena de conexión en appsettings.json para SQL Server.
3. Ejecutar migraciones de Entity Framework Core.
4. Levantar el backend con `dotnet run`.
5. Instalar dependencias del frontend con `npm install`.
6. Ejecutar el frontend con `ng serve`.

## **Link GitHub**

<https://github.com/josephcr117/sirec>

## **Referencias**

Bootstrap. (s.f.). Bootstrap: The most popular HTML, CSS, and JS library in the world. <https://getbootstrap.com/>

Flowchart maker & online diagram software. (s.f.). Draw.Io. <https://Draw.io>

GitHub. (s. f.). GitHub. <https://github.com/>

Microsoft. (2025). \*Microsoft SQL Server (v2022) [Software]. <https://www.microsoft.com/sql-server>

Microsoft. (2022). Visual Studio (v17.8.0) [Software]. <https://visualstudio.microsoft.com/>

Microsoft. (2023). Visual Studio Code (v1.85.1) [Software]. <https://code.visualstudio.com/>

B