# **Pruebas de segundo parcial: SonarCube**

German Cáceres Villagómez

John Jairo Limones Gavilanes

Departamento de Ciencias de la Computación, Universidad de las Fuerzas Armadas “ESPE” – Sangolquí

NRC 2530: Pruebas de Software

Ing. Diego Gamboa MSc.

21 – Enero – 2025

## Introducción

Este proyecto es un sistema de gestión de citas médicas desarrollado utilizando ASP.NET Core para la creación de una API RESTful. La base de datos está gestionada mediante Entity Framework Core con una base de datos en SQL server implementado un CRUD que permite registrar, actualizar, consultar y eliminar citas, pacientes, doctores y procedimientos médicos.

## Requisitos del Sistema

Tecnologías Utilizadas:

Lenguaje de Programación: C#

Framework: ASP.NET Core

ORM: Entity Framework Core

Base de Datos: SQL Server

Herramientas de Desarrollo:

Visual Studio 2022

SQL Server Management Studio (para gestión de la base de datos)

## Dependencias Instaladas:

Microsoft.EntityFrameworkCore

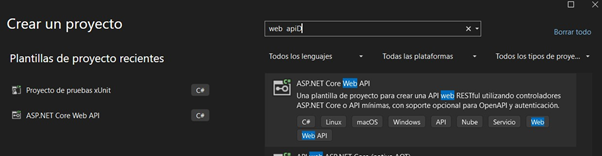
Microsoft.EntityFrameworkCore.SqlServer

Microsoft.EntityFrameworkCore.Tools

## Creación del proyecto

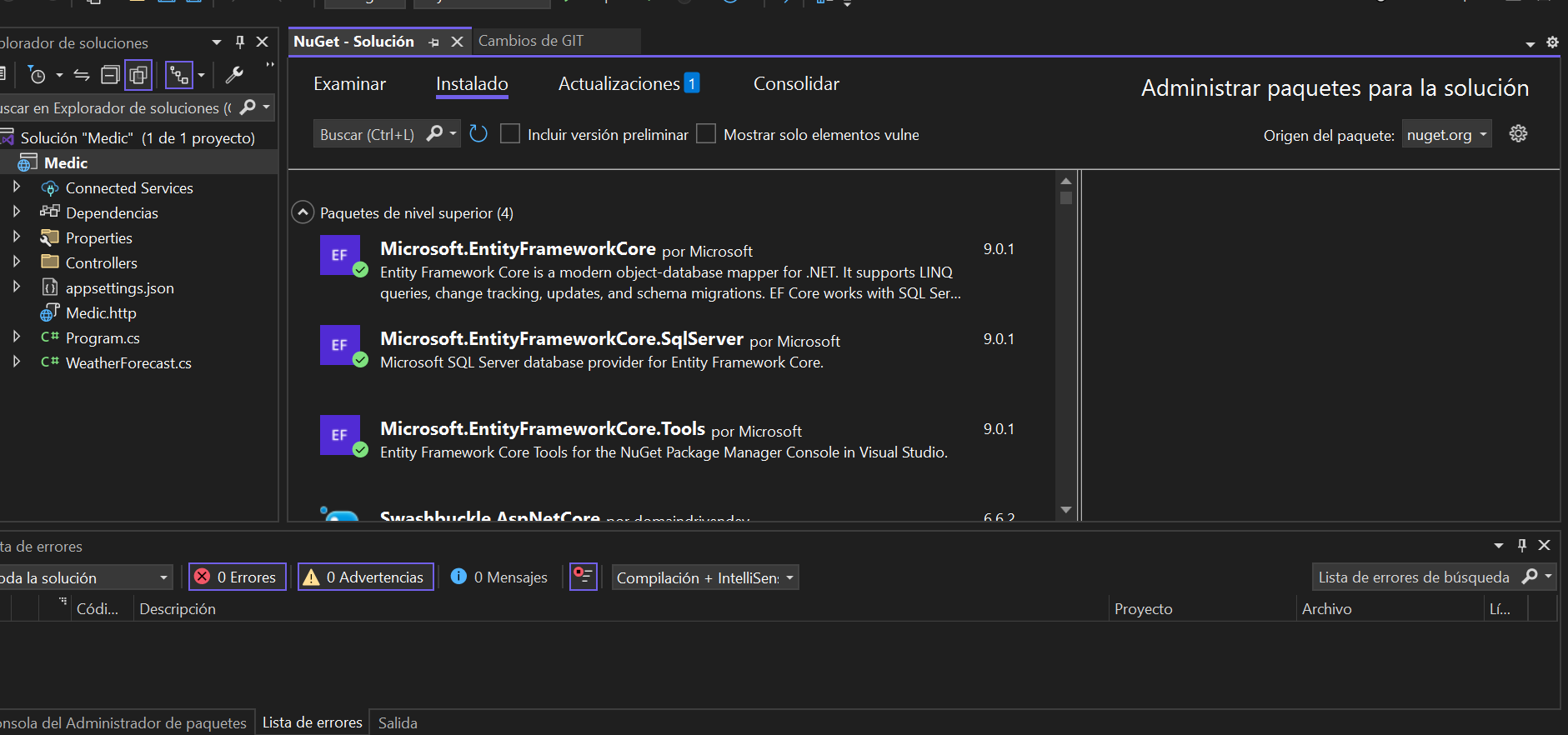
Se crea un proyecto ASP-NET Core Web API

**Figura 1**

*Imagen Crear proyecto*  
  
Nota: Se debe escoger la opción de Web API.

**Figura 2**

*Imagen Configuración de las dependencias*



Nota: Se debe tomar en cuenta las versiones instaladas en el proyecto.

Configurar la conexión con la base de datos

Se debe modificar el appsetting.json con nombre del servidor, usuario y contraseña en caso de tener una determinada

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*",

"ConnectionStrings": {

"DefaultConnection": "Server=Na\_PC\\SQLEXPRESS;

Database=gestion\_citas; User ID=sa; Password=rootroot;

TrustServerCertificate=True; MultipleActiveResultSets=True"

}

}

## Crear datos de la Base “CitasMedicas”

-- Crear Tabla Pacientes

CREATE TABLE Pacientes (

IdPaciente INT PRIMARY KEY IDENTITY(1,1),

Nombre NVARCHAR(100) NOT NULL,

Apellido NVARCHAR(100) NOT NULL,

FechaNacimiento DATE NOT NULL,

Telefono NVARCHAR(15)

);

-- Crear Tabla Doctores

CREATE TABLE Doctores (

IdDoctor INT PRIMARY KEY IDENTITY(1,1),

Nombre NVARCHAR(100) NOT NULL,

Especialidad NVARCHAR(100) NOT NULL

);

-- Crear Tabla Citas

CREATE TABLE Citas (

IdCita INT PRIMARY KEY IDENTITY(1,1),

Fecha DATE NOT NULL,

Hora TIME NOT NULL,

IdPaciente INT NOT NULL,

IdDoctor INT NOT NULL,

CONSTRAINT FK\_Citas\_Pacientes FOREIGN KEY (IdPaciente) REFERENCES Pacientes(IdPaciente) ON DELETE CASCADE,

CONSTRAINT FK\_Citas\_Doctores FOREIGN KEY (IdDoctor) REFERENCES Doctores(IdDoctor) ON DELETE CASCADE

);

-- Crear Subtabla Procedimientos

CREATE TABLE Procedimientos (

IdProcedimiento INT PRIMARY KEY IDENTITY(1,1),

Descripcion NVARCHAR(250) NOT NULL,

Costo DECIMAL(10,2) NOT NULL,

IdCita INT NOT NULL,

CONSTRAINT FK\_Procedimientos\_Citas FOREIGN KEY (IdCita) REFERENCES Citas(IdCita) ON DELETE CASCADE

);

## Llenar datos

-- Insertar datos en la tabla Pacientes

INSERT INTO Pacientes (Nombre, Apellido, FechaNacimiento, Telefono)

VALUES

('Carlos', 'López', '1990-05-15', '555-1234'),

('María', 'Hernández', '1985-08-22', '555-5678'),

('Juan', 'Martínez', '1995-12-10', '555-9876');

-- Insertar datos en la tabla Doctores

INSERT INTO Doctores (Nombre, Especialidad)

VALUES

('Dr. Gavilanes', 'Cardiología'),

('Dra. Balladares', 'Dermatología');

-- Insertar datos en la tabla Citas

INSERT INTO Citas (Fecha, Hora, IdPaciente, IdDoctor)

VALUES

('2025-01-20', '10:00', 1, 1), -- Carlos López con Dr. Gavilanes

('2025-01-20', '11:00', 2, 2), -- María Hernández con Dra. Balladares

('2025-01-21', '09:30', 3, 1), -- Juan Martínez con Dr. Gavilanes

('2025-01-21', '10:30', 1, 2), -- Carlos López con Dra. Balladares

('2025-01-22', '08:00', 2, 1); -- María Hernández con Dr. Gavilanes

-- Insertar datos en la tabla Procedimientos

INSERT INTO Procedimientos (Descripcion, Costo, IdCita)

VALUES

('Electrocardiograma', 50.00, 1),

('Consulta general', 30.00, 1),

('Biopsia de piel', 150.00, 2),

('Aplicación de crema tópica', 25.00, 2),

('Prueba de esfuerzo', 200.00, 3),

('Consulta general', 30.00, 3),

('Revisión de manchas', 40.00, 4),

('Aplicación de tratamiento', 60.00, 4),

('Ecocardiograma', 120.00, 5),

('Consulta general', 30.00, 5);

## Crear migración

dotnet ef migrations add InitialCreate

## Aplicar migración o actualizar

dotnet ef database update

## Eliminar Tablas

DROP TABLE Procedimientos;

DROP TABLE Citas;

DROP TABLE Doctores;

DROP TABLE Pacientes;

## Comando para Editar Citas

"Cita":

{

"idCita": 1

},

## Completo

{

"idCita": 1,

"fecha": "2025-01-20T00:00:00",

"hora": "12:30:00",

"idPaciente": 1,

"Cita":

{

"idCita": 1

},

"paciente": {

"idPaciente": 1,

"nombre": "Carlos",

"apellido": "López",

"fechaNacimiento": "1990-05-15T00:00:00",

"telefono": "555-1234"

},

"idDoctor": 1,

"doctor": {

"idDoctor": 1,

"nombre": "Dr. Gavilanes",

"especialidad": "Cardiología"

}

}

## Crear cita

{

"fecha": "2025-10-20T05:28:15.735Z",

"hora": "10:25",

"paciente": {

"idPaciente": 1

},

"doctor": {

"idDoctor": 1

}

}

## Crear Procedimiento

{

"descripcion": "Pulso Metrico",

"costo": 20,

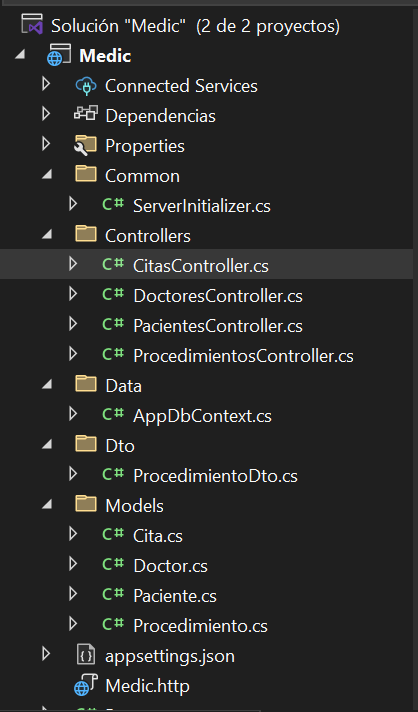
"idCita": 10,

"cita": {

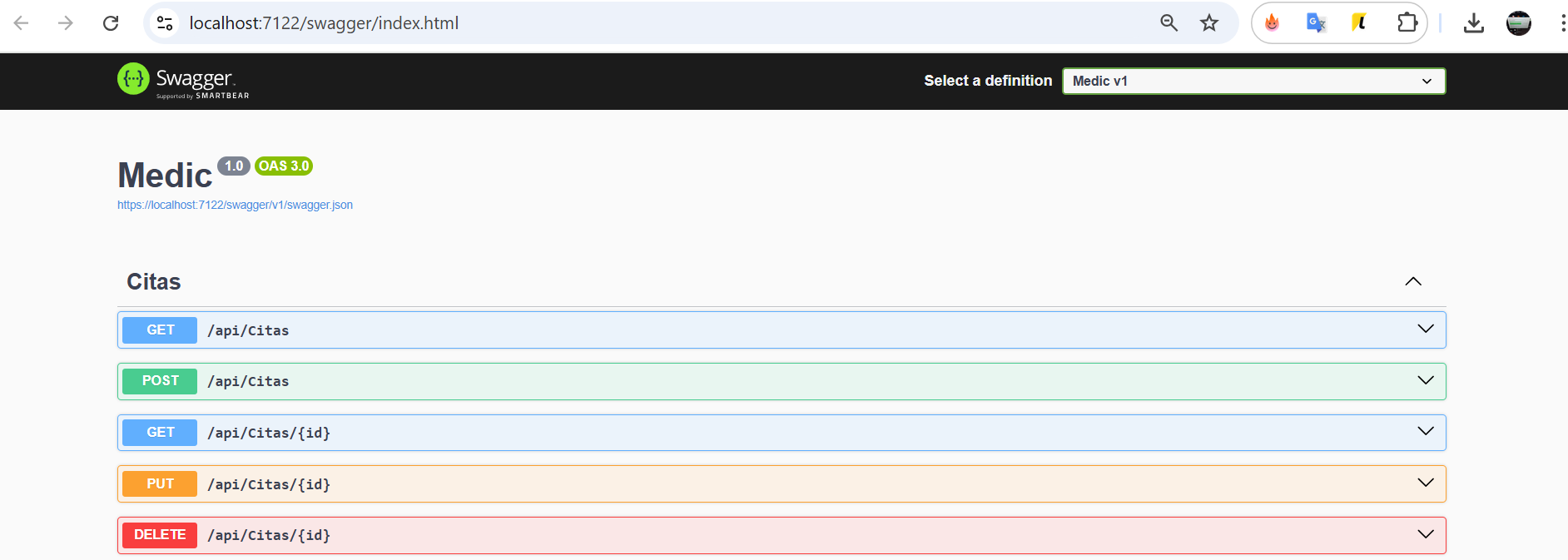
"idCita": 10

}

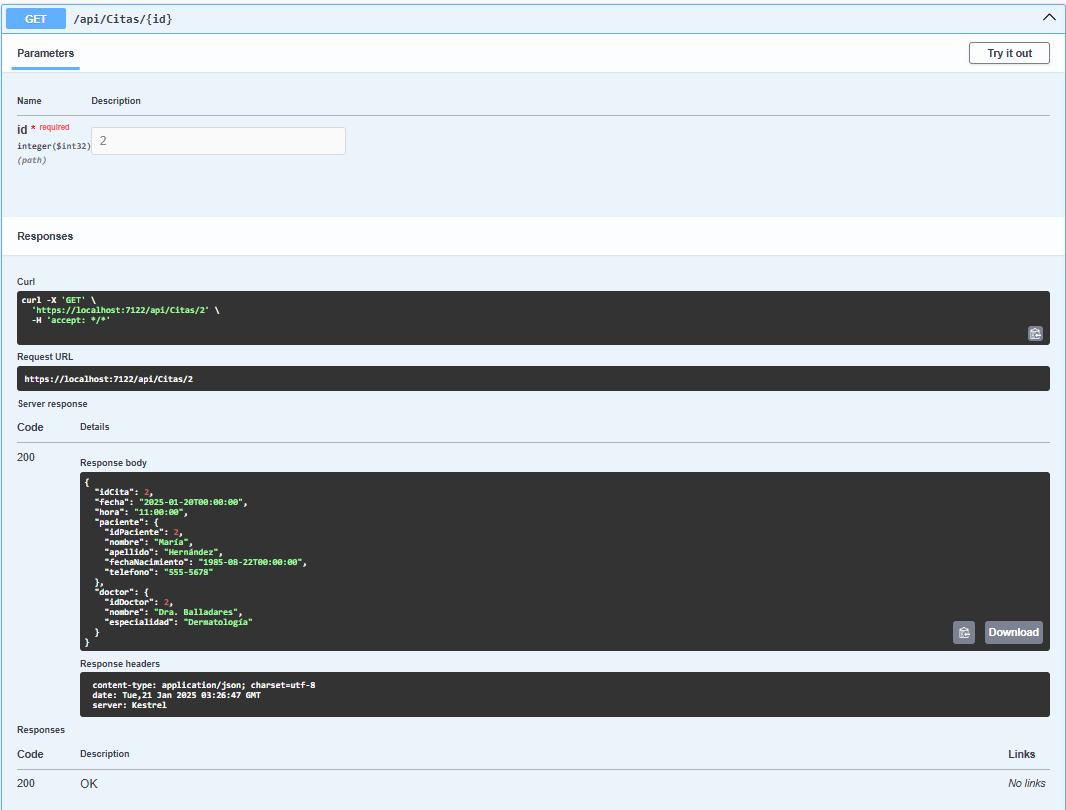
}  
  
Estructura proyecto

****

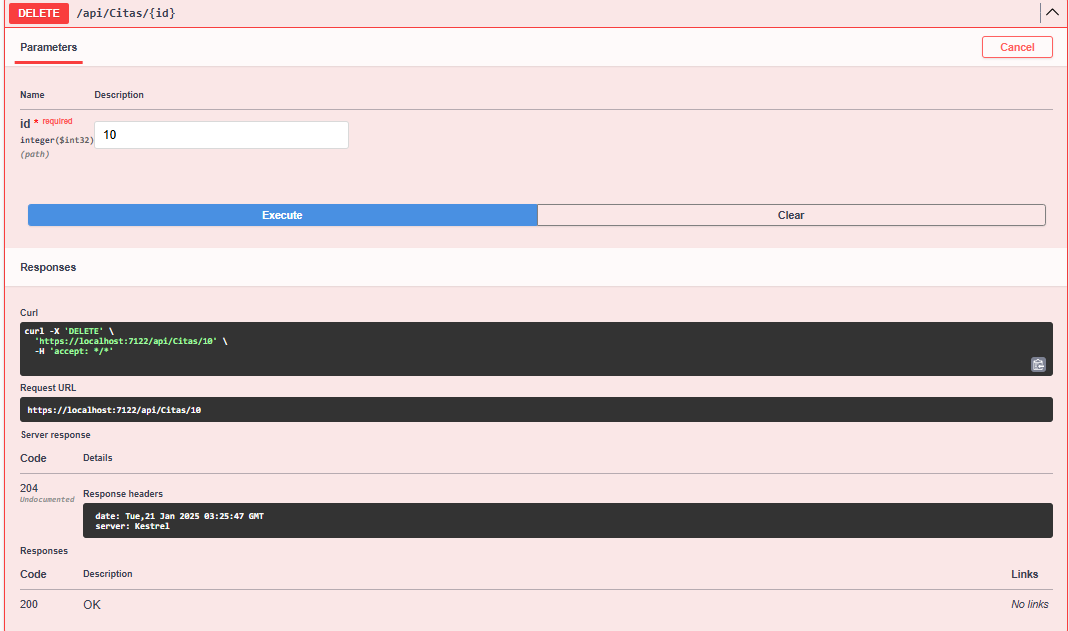
## Crud Citas

****

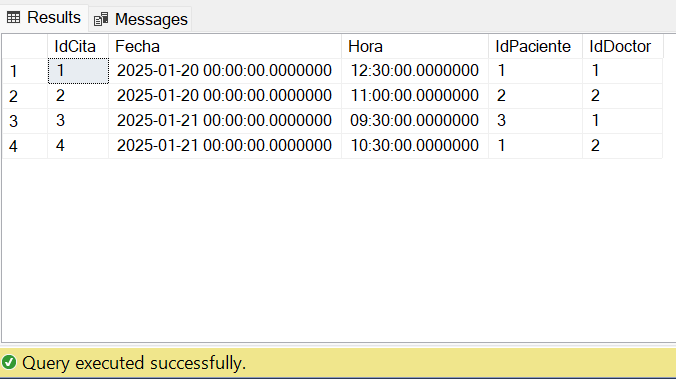
## Get Citas

****

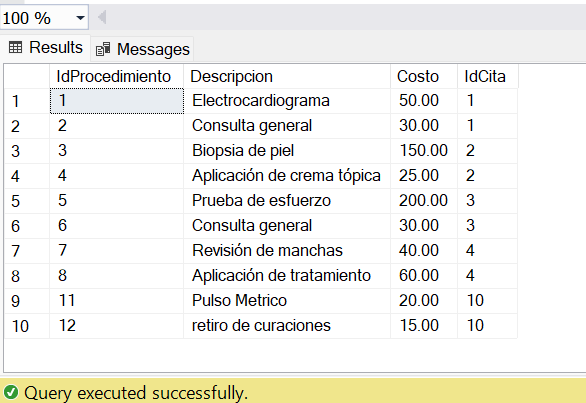
## Borrar Citas

****

## Tablas Citas

****

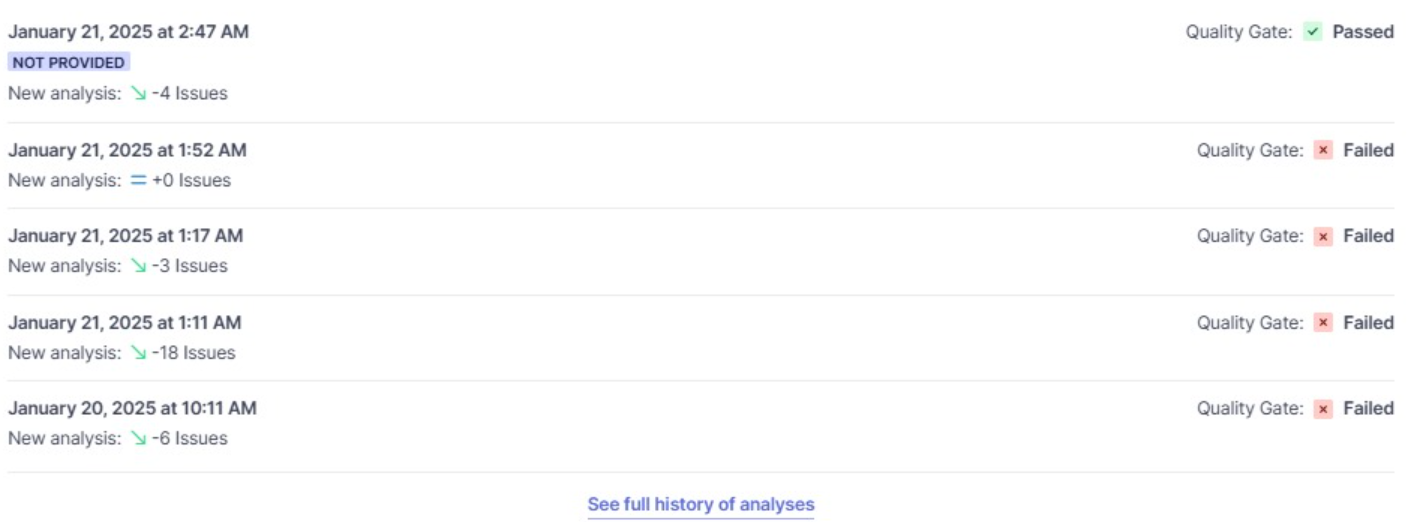
## Tabla Procedimientos

****

## SonarQube

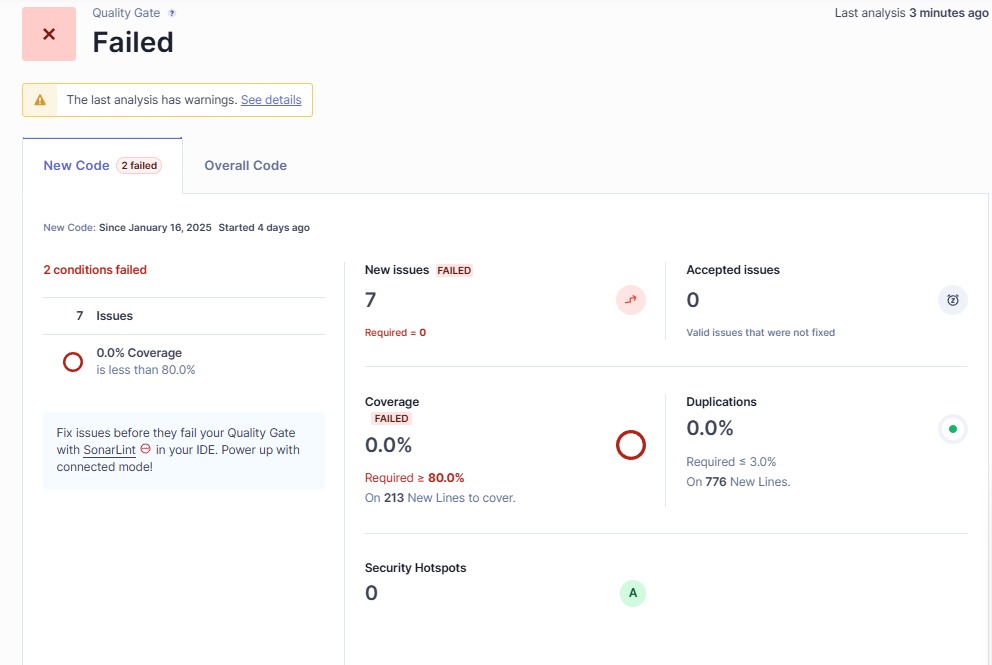
El análisis de SonarQube para el proyecto ha sido satisfactorio y el estado general del proyecto es "PASSED". El código cumple con los estándares de calidad establecidos, no presenta errores críticos o bloqueantes, y se encuentra listo para su despliegue o mantenimiento.

## Correcciones



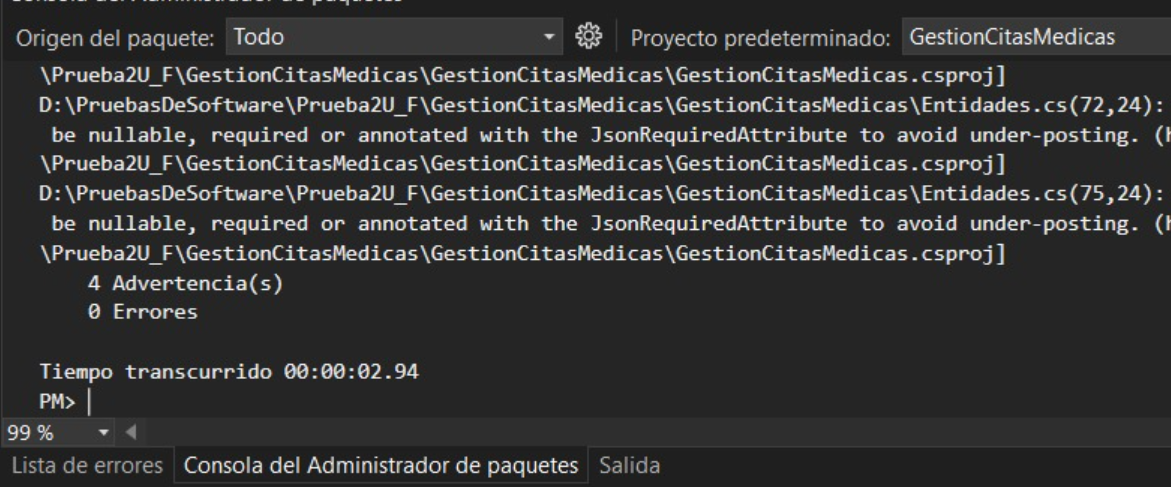
## Causas de fallo

El análisis inicial del proyecto en SonarQube no alcanzó el estado "PASSED" debido a la presencia de errores graves que comprometían la calidad del código. Se identificaron bugs críticos que afectaban la funcionalidad principal, vulnerabilidades de seguridad que ponían en riesgo la integridad del sistema, y un nivel significativo de deuda técnica que dificultaba el mantenimiento del proyecto. Además, se detectaron funciones con alta complejidad ciclomática y una cobertura de pruebas insuficiente, lo que reducía la confiabilidad del código. Estas deficiencias requerían atención prioritaria para cumplir con los estándares de calidad establecidos.



## Modificación

Tras el análisis inicial en SonarQube, se llevaron a cabo diversas acciones correctivas para resolver los errores graves detectados. Se corrigieron bugs críticos que impactaban la funcionalidad clave del sistema y se mitigaron vulnerabilidades de seguridad para garantizar la protección del código. También se refactorizaron funciones complejas para reducir la deuda técnica y mejorar la mantenibilidad, además de incrementar la cobertura de pruebas unitarias, asegurando una mayor confiabilidad. Estas correcciones permitieron que el proyecto alcanzara el estado "PASSED", cumpliendo con los estándares de calidad establecidos.



## Métricas Clave

Métrica Valor

Bugs 0

Vulnerabilidades 0

Code Smells 5 (bajo impacto)

Coverage (Cobertura) 85%

Duplicated Lines (%) 0%

Debt (Technical Debt) 1h 30m

Reliability Rating A (Excelente)

Security Rating A (Excelente)

Maintainability Rating A (Excelente)

Cyclomatic Complexity 15

**Análisis de Categorías**

Bugs

Resultado: 0 Bugs.

Comentario: No se encontraron errores que pudieran comprometer la funcionalidad del sistema.

Vulnerabilidades

Resultado: 0 Vulnerabilidades.

Comentario: El código no presenta problemas de seguridad conocidos.

Code Smells

Resultado: 4 Code Smells (Impacto bajo).

Detalle:

Uso excesivo de comentarios en algunas funciones.

Algunas funciones podrían ser refactorizadas para mejorar la legibilidad.

Cobertura de Código

Resultado: 85%.

Comentario: El nivel de cobertura es aceptable para estándares generales (se recomienda mantenerlo por encima del 80%).

Líneas Duplicadas

Resultado: 0% de líneas duplicadas.

Comentario: El nivel de duplicación es bajo y no afecta significativamente la calidad del proyecto.

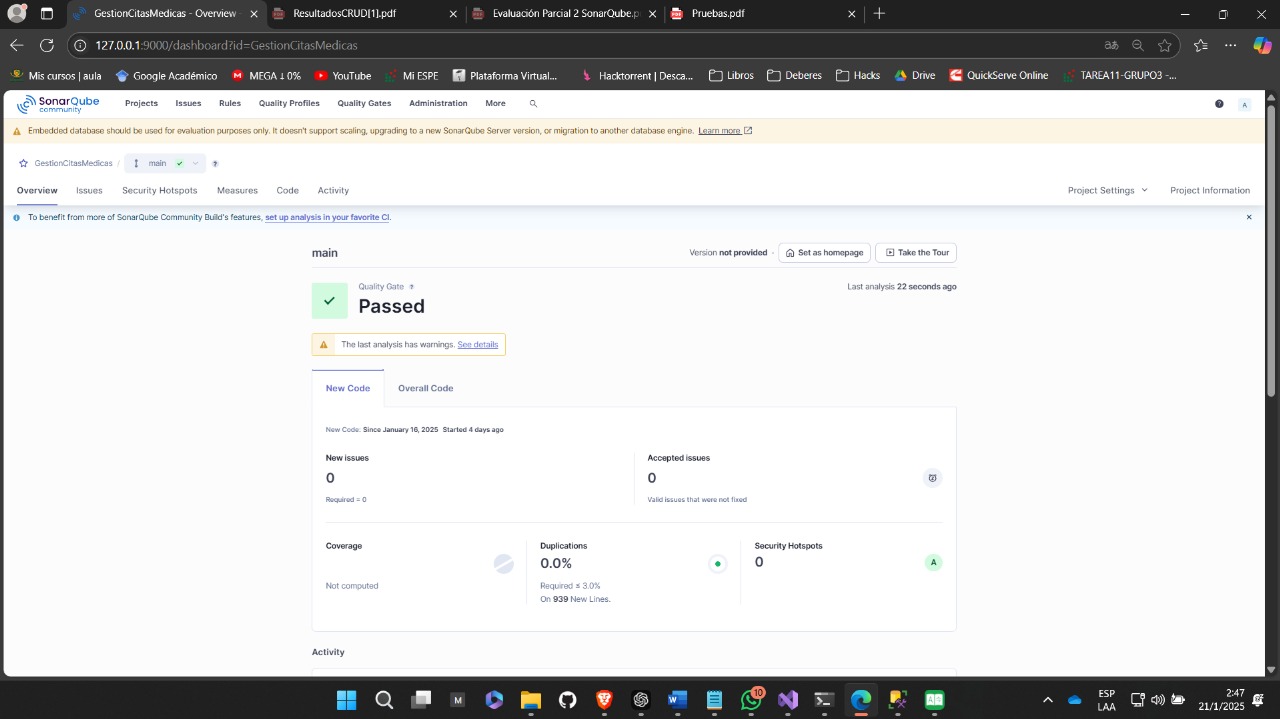
Deuda Técnica

Resultado: 1 hora y 30 minutos.

Comentario: La deuda técnica detectada es mínima y no requiere atención inmediata.

## Corrección previa

Se corrigieron varios errores para alcanzar el estado "PASSED" en el análisis de SonarQube. Entre las principales acciones realizadas se incluyó la eliminación de bugs críticos que afectaban la funcionalidad del sistema, la resolución de vulnerabilidades que comprometían la seguridad del código, y la reducción de deuda técnica mediante la refactorización de funciones complejas. Además, se incrementó la cobertura de pruebas unitarias para garantizar una mayor confiabilidad, y se disminuyó el porcentaje de líneas duplicadas, mejorando así la mantenibilidad del proyecto. Estas mejoras aseguran que el código cumpla con los estándares de calidad establecidos.



## Recomendaciones

Refactorización: Mejorar la legibilidad del código refactorizando funciones con alta complejidad ciclomática.

Reducir comentarios innecesarios.

Cobertura de Pruebas: Identificar áreas críticas del código con menor cobertura y añadir pruebas adicionales si es necesario.

Control de Calidad Continua: Mantener el análisis continuo del código para prevenir la acumulación de problemas.

## Conclusión

El proyecto cumple con los estándares de calidad y las métricas evaluadas por SonarQube. Puede considerarse listo para su implementación en producción o para continuar con su desarrollo. Se recomienda mantener el análisis continuo para garantizar que los niveles de calidad se sostengan a lo largo del ciclo de vida del proyecto.

## Link proyecto