

# PROYECTO DE ANALISIS Y MEJORAMIENTO DE SOFTWARE

Katerin Merino Quispe (2018060918), Percy Taquila Carazas (2018061088),  
Abraham Lipa Calabilla (2019064039), Edwart (), Liz ()

*Tacna, Perú*

---

## Abstract

This document is intended to present patient care in a concise and clear way, diagnosing their possible disease with terms of software that would be carried out in the system. In this documentation, the requirements that will serve as a guide to develop the software in its different stages will be reflected, helping us to validate and inspect its construction, applying the Software quality. Therefore, we will work with Sonarqube for the analysis of static code to analyze the code and find code errors, and security vulnerabilities. SonarSource's C Sharp analysis has extensive coverage of well-established quality standards.

---

## Resumen

Con el presente documento se pretende presentar de forma concisa y clara las necesidades del cliente en el área de ventas en términos de software que se realizará. En esta documentación se plasmará los requerimientos que servirán de guía para desarrollar el software en sus distintas etapas, ayudándonos a validar e inspeccionar la construcción de este, aplicando la calidad de Software. Por ello se trabajara con Sonarqube para el análisis de código estático para analizar el código y encontrar errores de código, y vulnerabilidades de seguridad. El análisis C-Sharp de SonarSource tiene una gran cobertura de estándares de calidad bien establecidos.

## **1. Antecedentes o Introducción**

El Sistema de Diagnostico Médico está desarrollado para que la atención a los pacientes sea mas rápida, ya que en esta pandemia hace que la atencion sea insufiente, por falta de tiempo y con este sistema se busca una atencion mas rápido y de forma ordenada, a los pacientes que vienen para hacer sus consultas. Con este sistema se diagnostica al paciente con los síntomas que nos diga en el momento de la consulta y dar como resultado la posible enfermedad que padece.

## **2. Título**

”Sistema de Diagnostico Medico”

Propuesta de un sistema de atención al paciente diagnosticando su posible enfermedad.

## **3. Autores**

- Katerin Merino Quispe
- Abraham Lipa Calabilla
- Percy Taquila Carazas

## **4. Planteamiento del problema**

### **4.1. Problema**

Falta e insuficiencia de herramientas para el apoyo en el diagnóstico medico de enfermedades, además por la pandemia que está pasando el mundo entero la demanda de pacientes no está siendo cubierta. Es por ello que muchos pacientes no son atendidos y por esta razon se necesita un sistema que apoye en el diagnostico, para poder agilizar la atención hacia los pacientes.

### **4.2. Justificación**

Crear un sistema de diagnóstico para las enfermedades es necesario debido a que puede ayudar al medico a brindar una respuesta más rápida, la implementación de este sistema va a mejorar la gestión de los servicios y la atención.

#### 4.3. *Alcance*

El software a desarrollar servira de apoyo para poder diagnosticar algunas enfermedades de acuerdo a los sintomas que el paciente presente.

### 5. **Objetivos**

#### 5.1. *General*

- Desarrollar un sistema de diagnóstico médico para optimizar el tiempo de respuesta del diagnóstico médico.

#### 5.2. *Específicos*

- Que el diagnóstico médico muestre las posibles enfermedades de un paciente.
- Detectar que tipo de posible enfermedad tiene el paciente.
- Almacenar el historial del paciente en la base de datos.

### 6. **Referentes teóricos**

### 7. **Desarrollo de la propuesta**

La calidad de código suele decirse que es un atributo interno de calidad, dado que no se hace visible al usuario. Pero llega un momento en el cual este atributo de calidad pasa de ser interno a externo, y esto se da cuando el hecho de tener modificar el código para hacer un cambio lleva mucho más tiempo del que debería. Con el fin de verificar la calidad interna de un sistema se suelen hacer análisis de código con SonarQube o herramientas similares. En este documento se muestra parte de de nuestro proyecto , donde básicamente cuenta cómo hacer una prueba de concepto rápidamente usando una imagen Docker de SonarQube, y ejecutando el análisis desde SonarQube Scanner. SonarQube, como tantas otras herramientas similares, permite realizar análisis estático de código fuente de manera automática, buscando patrones con errores, malas prácticas o incidentes. Además, realiza un cálculo de la deuda técnica. Dentro de las verificaciones que hacen herramientas como SonarQube, se encuentran las siguientes:

- Detección de código duplicado..
- Falta de pruebas unitarias, falta de comentarios.

- Código spaghetti, complejidad ciclomática, alto acoplamiento.
- Tamaño de archivos de código.
- Tamaño de métodos.
- No adecuación a estándares y convenciones de código.
- Vulnerabilidades conocidas de seguridad.

Una muestra en Nuestro proyecto:

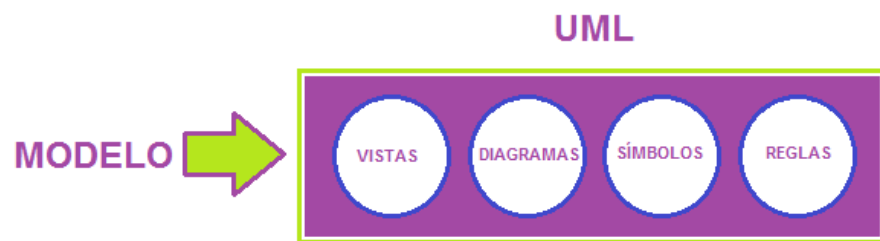
	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
SistemaDiagnostico.Datos.EF	292	0	0	10	0	0.0%	28.5%
SistemaDiagnostico.Negocio	233	0	0	38	0	0.0%	0.0%
SistemaDiagnostico.Presentacion	2,039	13	0	156	0	0.0%	30.1%
SistemaDiagnostico.PruebasUnitarias	—	0	0	0	0	—	—

### 7.1. *Tecnología de información*

- SQL SERVER: Microsoft SQL Server es un sistema de gestión de base de datos relacional, desarrollado por la empresa Microsoft. El lenguaje de desarrollo utilizado es Transact-SQL, una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos, crear tablas y definir relaciones entre ellas..
- C-Sharp : lenguaje de programación multiparadigma desarrollado y estandarizado por Microsoft. Es un lenguaje de programación creado para diseñar aplicaciones en la plataforma.NET.
- Visual Studio: es un entorno de desarrollo en diferentes sistemas operativos y compatibles con múltiples lenguajes de programación al igual que entornos de desarrollo web.

## 7.2. Metodología, técnicas usadas

UML es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Existen diferencias importantes entre un método y un lenguaje de modelado. Un método es una manera explícita de estructurar el pensamiento y las acciones de cada individuo. Además, el método le dice al usuario qué hacer, cómo hacerlo, cuándo hacerlo y por qué hacerlo; mientras que el lenguaje de modelado carece de estas instrucciones. Los métodos contienen modelos y esos modelos son utilizados para describir algo y comunicar los resultados del uso del método.



## 8. Cronograma

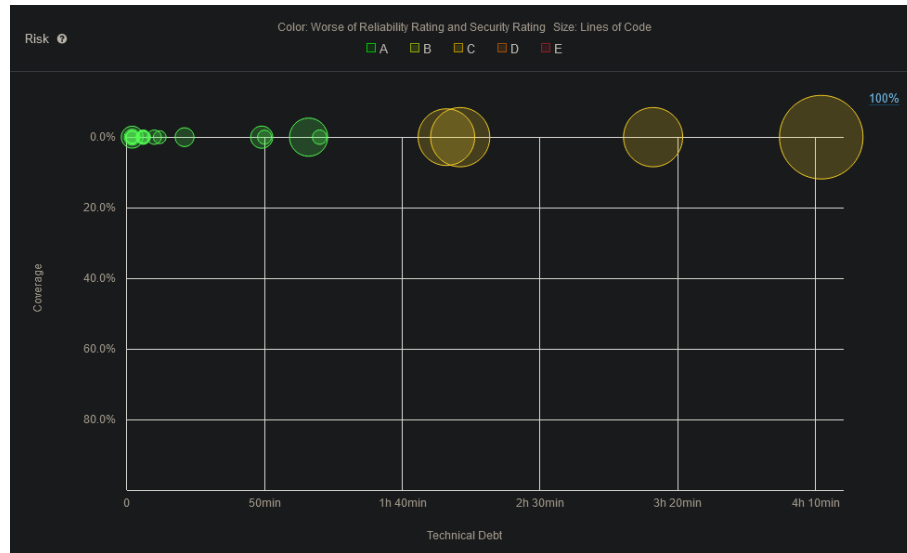
The screenshot shows the Azure DevOps interface for the repository "ProyectoSistemaDiagnosticoMedico". The left sidebar contains navigation options: Overview, Boards, Repos, Files, Commits, Pushes, Branches, Tags, Pull requests, Pipelines, Test Plans, Artifacts, and Project settings. The "Files" view is selected, showing a tree structure of the repository. The "Contents" tab is active, displaying a table of files and folders.

Name	Last change	Commits
.vs	8 oct.	db073bfb subiendo readme P-TaquillaCc
SistemaDiagnostico.	8 oct.	db073bfb subiendo readme P-TaquillaCc
SistemaDiagnostico.	8 oct.	db073bfb subiendo readme P-TaquillaCc
SistemaDiagnostico.	8 oct.	db073bfb subiendo readme P-TaquillaCc
SistemaDiagnostico.	8 oct.	db073bfb subiendo readme P-TaquillaCc
SistemaDiagnostico.A	5 oct.	35f19a43 first commit P-TaquillaCc
1. Script_BD_Tabla...	5 oct.	35f19a43 first commit P-TaquillaCc
2. Procedimient...	8 oct.	db073bfb subiendo readme P-TaquillaCc
ProyectoDiagnosto...	5 oct.	35f19a43 first commit P-TaquillaCc
MI README.md	8 oct.	db073bfb subiendo readme P-TaquillaCc
w32pi661.exe	5 oct.	35f19a43 first commit P-TaquillaCc

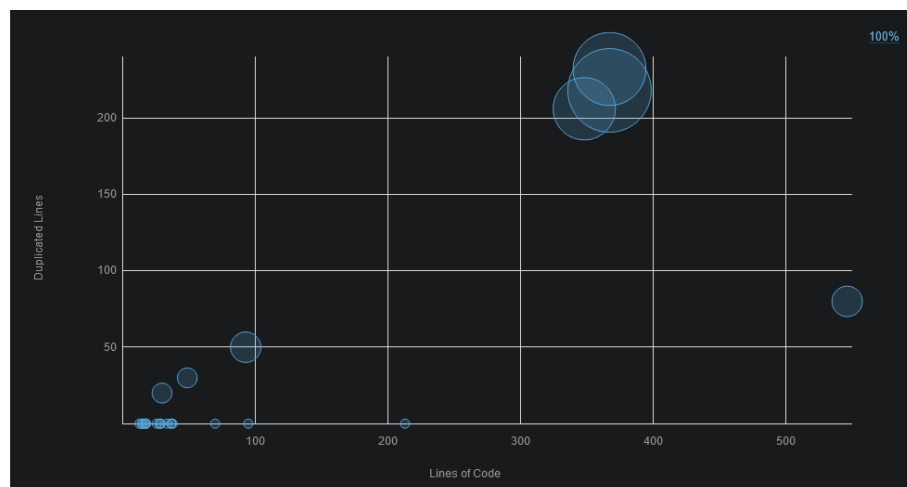
## 9. Resultado Sonarqube

### Definición de métricas

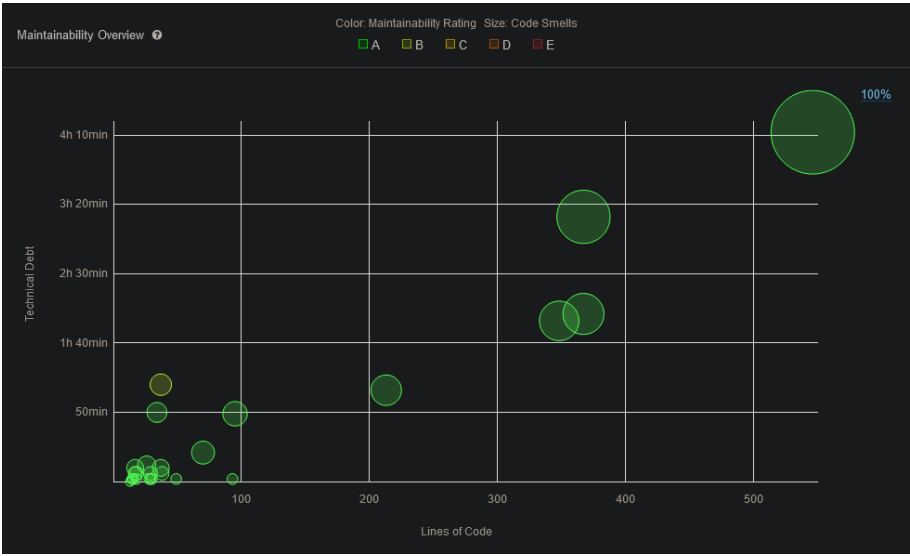
- Descripción del proyecto



- Código duplicado

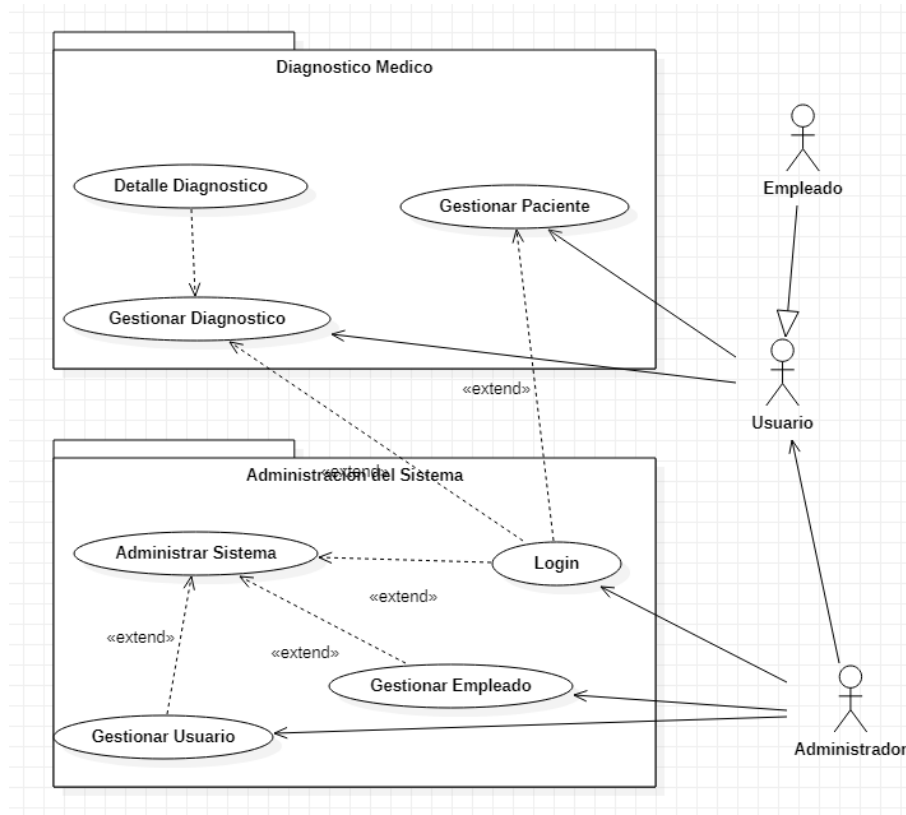


- Mantenibilidad



## 10. Desarrollo de Solución de Mejora

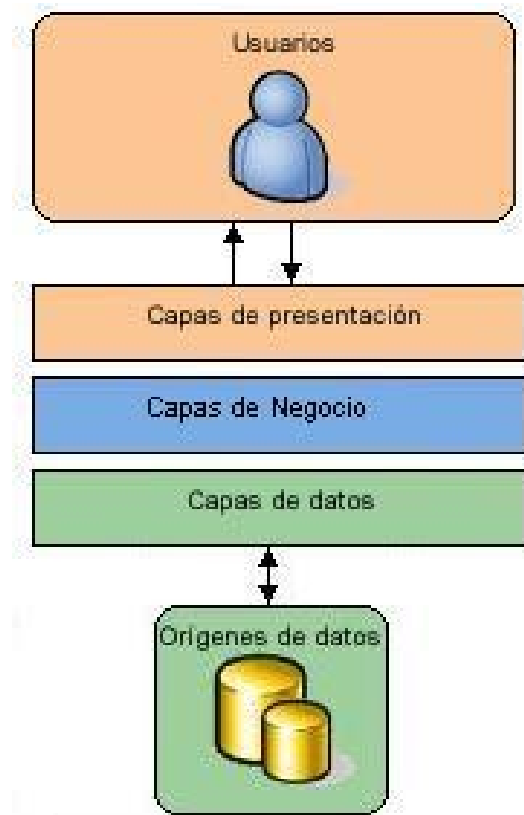
### 10.1. Casos de Uso de la aplicación

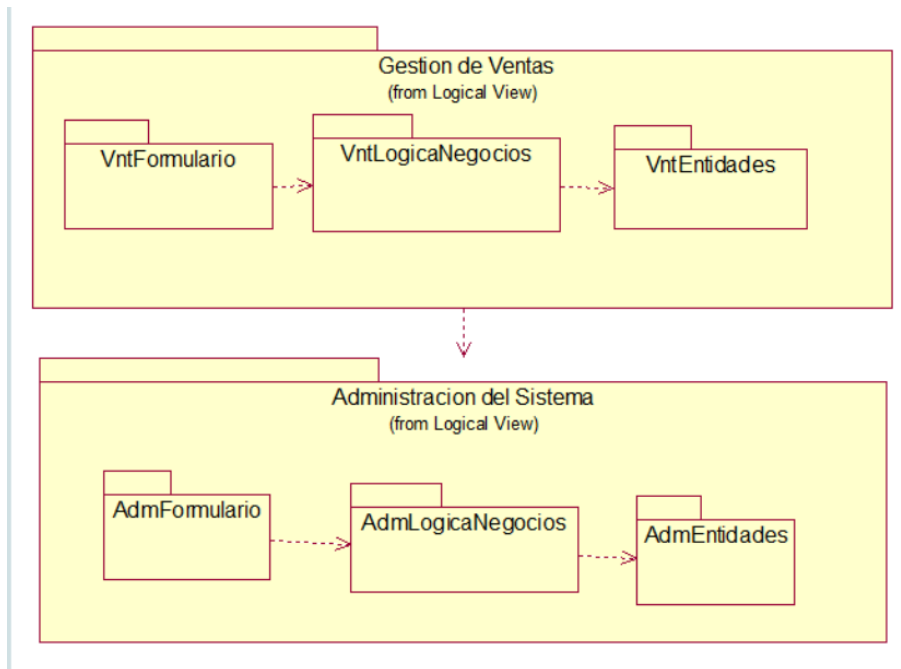


### 10.2. Diagrama de Arquitectura de la aplicación

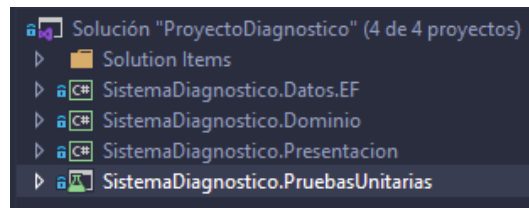
- Existen diferentes tipos de arquitectura o patrones a seguir para desarrollar un software, en este caso voy a explicar en que consiste la arquitectura de 3 Capas , que ami parecer es la mas general o la mas basica para desarrollar.
- La Capa de Presentación: Donde se encuentran los formularios y la parte visual de la aplicación.
- La Capa de Negocios: Donde se encuentra toda la logica del negocio y clases que las componete es decir, Entidades y controladoras
- La Capa de Acceso a Datos: Donde se encuentra las conexiones y las transacciones que se utilizan para comunicarse con la base de datos.



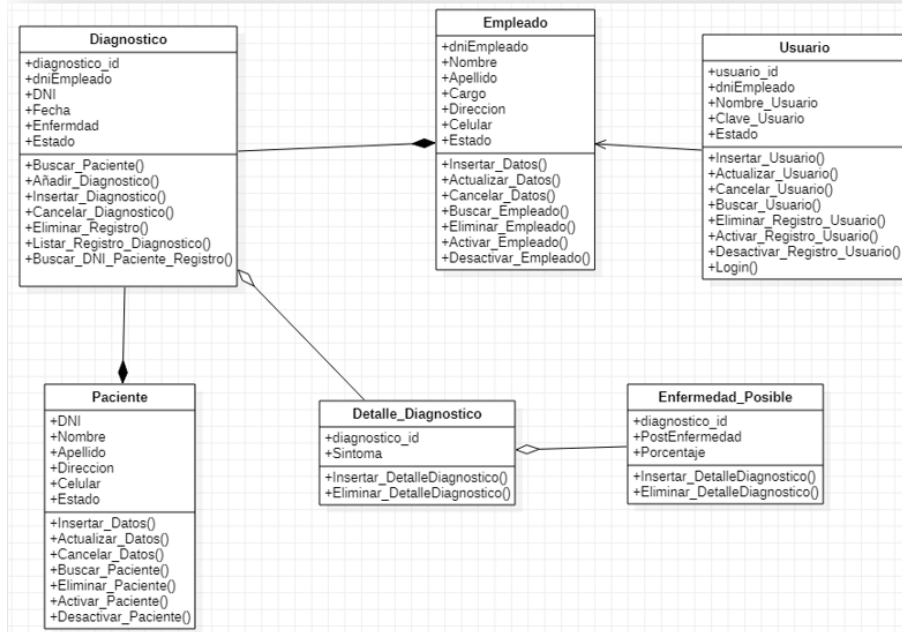




Estructura de la solución en Visual Studio



### 10.3. Diagrama de Clases de la aplicación



### 10.4. Metodos de pruebas implementados para coberturar la aplicación

#### 1 Pruebas Unitarias

```

11 public void CU01_Insert_AreEqualTrue_IfAddAndFind()
12 {
13     Empleados empleado = new Empleados
14     {
15         Dni = "76368626",
16         Nombre = "Abraham",
17         Apellido = "Lipa",
18         Direccion = "Av. Leguia 123",
19         Celular = "942917242",
20         Cargo = "Medico",
21         Estado = "A"
22     };
23
24     using (var db = new SDiagnostico())
25     {
26         db.Empleados.Add(empleado);
27         db.SaveChanges();
28
29         var dniBuscado = db.Empleados.Where(e => e.Dni == "76368626").First().Dni;
30         Assert.AreEqual(empleado.Dni, dniBuscado);
31     }
32 }

```

```

34 public void CU02_Update_ShouldTrue_IfFindAndFind()
35 {
36     using (var db = new SDiagnostico())
37     {
38         var empleado = db.Empleados.Where(e => e.Dni == "76368626").First();
39         empleado.Apellido = "Lipa Calabilla";
40         db.Empleados.Add(empleado);
41
42         db.Entry(empleado).State = System.Data.EntityState.Modified;
43         db.SaveChanges();
44     }
45
46     using (var db = new SDiagnostico())
47     {
48         var empleadoBuscado = db.Empleados.Where(e => e.Dni == "76368626").First();
49         var apellidoBuscado = empleadoBuscado.Apellido;
50         Assert.AreEqual("Lipa Calabilla", apellidoBuscado);
51     }
52 }







```

```

54 public void CU03_Delete_ShouldTrue_IfRemoveAndFind()
55 {
56     using (var db = new SDiagnostico())
57     {
58         var empleado = db.Empleados.Where(e => e.Dni == "76368626").First();
59
60         db.Empleados.Remove(empleado);
61         db.SaveChanges();
62
63         Assert.ThrowsException<InvalidOperationException>(() => db.Empleados
64             .Where(e => e.Dni == "76368626").First());
65     }
66 }

```

## 2 Ejecución de pruebas unitarias

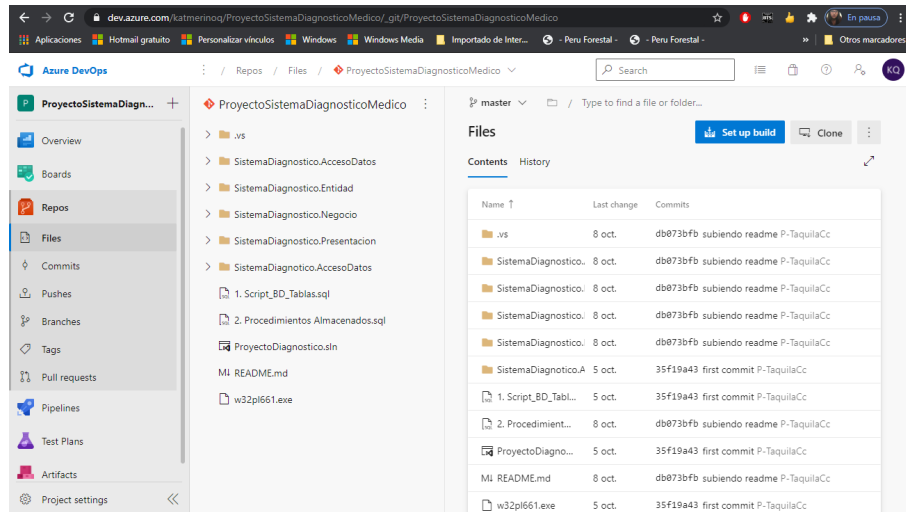
Prueba	Duración
▲  SistemaDiagnostico.PruebasUnitarias (3)	2 s
▲  SistemaDiagnostico.PruebasUnitarias (3)	2 s
▲  PruebasEF (3)	2 s
 CU01_Insert_AreEqualTrue_IfAddAndFind	2 s
 CU02_Update_ShouldTrue_IfFindAndFind	149 ms
 CU03_Delete_ShouldTrue_IfRemoveAndFind	21 ms

## 3 Pruebas basadas en Desarrollo Guiado por el Comportamiento

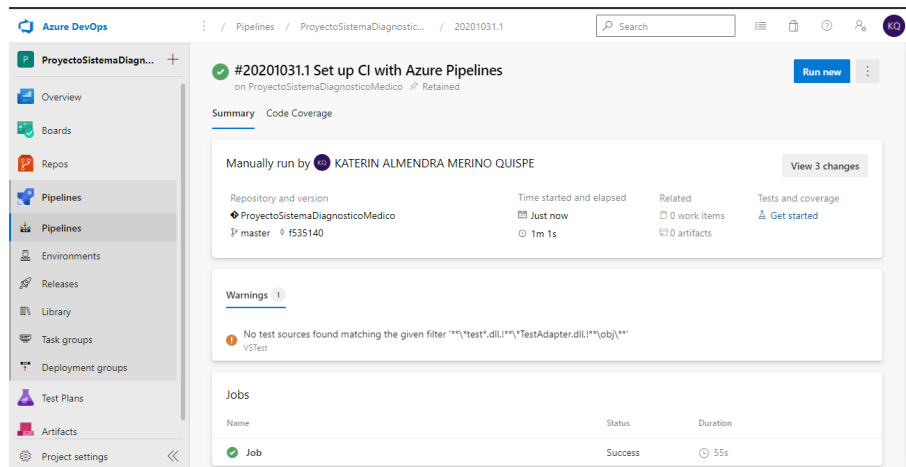
### 11. Desplegar las pruebas automatizadas Azure Devops y Git gub actions

Para hacer la prueba se clonado desde el repositorio de GitHub el proyecto

- Para hacer la prueba se clonado desde el repositorio de GitHub el proyecto



- se crea Pipelines



- y posteriorme a la ejecucion

