

PROYECTO DE ANALISIS Y MEJORAMIENTO DE SOFTWARE

Katerin Merino Quispe (2018060918), Percy Taquila Carazas (2018061088),
Abraham Lipa Calabilla (2019064039), Edwart Balcon Coahila
(2013046516), 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

Este documento tiene como objetivo presentar la atención al paciente de forma concisa y clara, diagnosticando su posible enfermedad con términos de software que se llevarían a cabo en el sistema. En esta documentación se reflejarán los requisitos que nos servirán de guía para desarrollar el software en sus diferentes etapas, ayudándonos a validar e inspeccionar su construcción, aplicando la calidad del Software.

Por lo tanto, trabajaremos 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 amplia 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 Médico”

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

3. Autores

- Katerin Merino Quispe
- Abraham Lipa Calabilla
- Percy Taquila Carazas
- Edwart Balcon Coahila

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**

6.1. *Título:*

Sistema Experto para el Diagnóstico de Enfermedades Respiratorias en el Hospital Central de la Policía Nacional del Perú Luis N. Sáenz

6.2. *Autor(es):*

Harold Anderson Chacaltana La Rosa

6.3. *Año:*

2017

6.4. *Resumen:*

El objetivo principal del presente trabajo es crear un sistema experto para el diagnóstico de enfermedades respiratorias, el cual será utilizado inicialmente en el Área de Neumología del Hospital Central PNP Luis N. Sáenz; permitiendo con ello atender de manera inmediata las inquietudes de los pacientes. El principal problema que afronta la institución es la carencia de personal médico especializado en el rubro de neumología, para atender a más de 500 pacientes al día, tal situación se complica aún más por la carencia de un sistema informático para el apoyo del diagnóstico.

El presente trabajo de investigación se realiza con la finalidad de aminorar los tiempos de respuesta del diagnóstico de las enfermedades respiratorias; debido a la afluencia de pacientes los médicos no se dan abasto en la atención, demorando en analizar los síntomas, por lo tanto, el tiempo de consulta es mayor; como consecuencia la atención es más prolongada de lo esperado, generando la reducción de pacientes a atender.

En el área de neumología debido al clima, la cantidad de incidentes va en aumento, a ello se suma la contaminación ambiental, y el poco personal especializado para la atención. [...]

6.5. *Título:*

Sistema Experto de Diagnostico Médico del Síndrome de Guillian Barre

6.6. *Autor(es):*

Marlene Carlos Soto

6.7. *Año:*

2002

6.8. *Resumen:*

[...] De esta manera esta monografía de Tesis tratará en el Capítulo 1: Teoría de Sistemas Expertos lo concerniente al fundamento teórico de los Sistemas Expertos: historia, definición, razones para la utilización de un Sistema Experto, los componentes que tiene, las áreas de aplicación de estos Sistemas y los tipos en los que se han clasificado estos sistemas. También se explicará la metodología existente para la confección de Sistemas Expertos.

En el Capítulo II: Técnicas Aplicadas a Sistemas Expertos, se verá como se representa el conocimiento para la implementación de un sistema experto,

así como los métodos de resolución de problemas para las formas de representación más conocidas. En este capítulo se explicarán las Redes Neuronales, conceptos, Historia, Clasificación, Aplicaciones y el Algoritmo de Diagnóstico utilizando las Redes Neuronales.

En el Capítulo III: Síndrome de Guillian Barre y su diagnóstico se desarrollará el tema de aplicación de esta tesis, antes de ello se explicará como el médico llega a un diagnóstico y luego se verá la enfermedad, síntomas y su diagnóstico.

En el Capítulo IV: Un Sistema Experto para el diagnóstico del Síndrome Guillian Barre se explicará la aplicación realizada, la base de conocimiento, y el algoritmo utilizado para la solución del problema. [...]

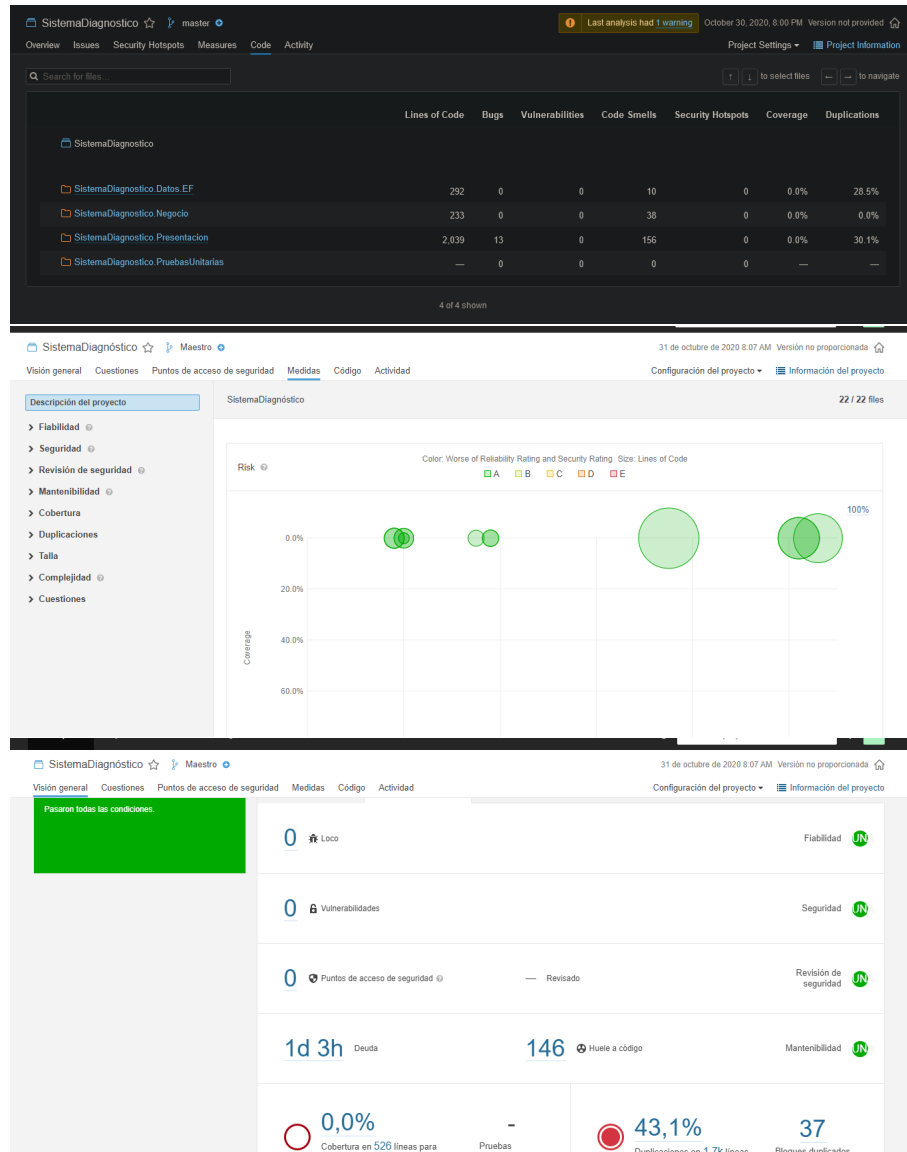
7. Desarrollo de la propuesta

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.

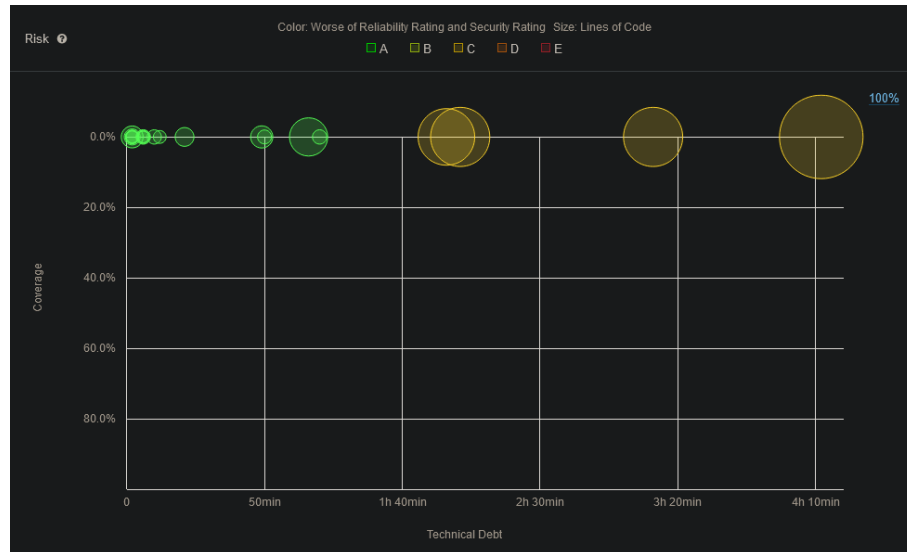
Una muestra en Nuestro proyecto:



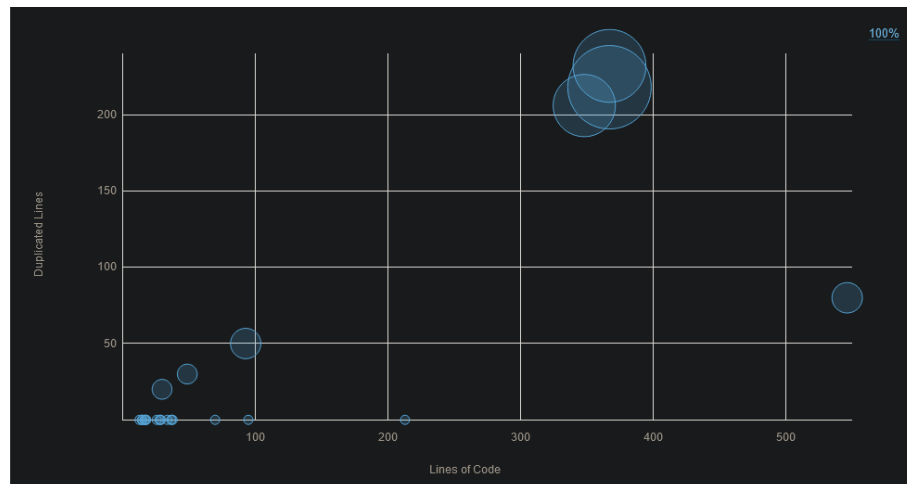
8. Resultados Sonarqube

Definición de métricas

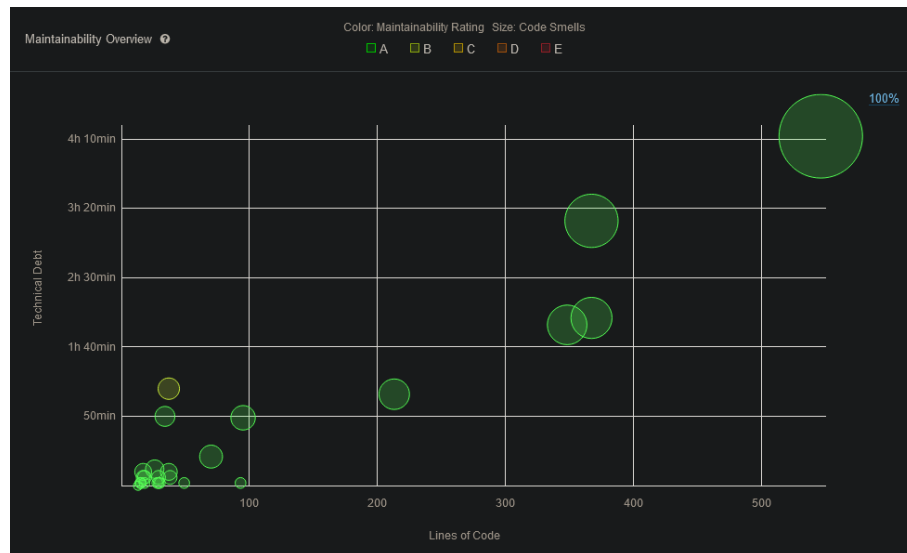
- Descripción del proyecto



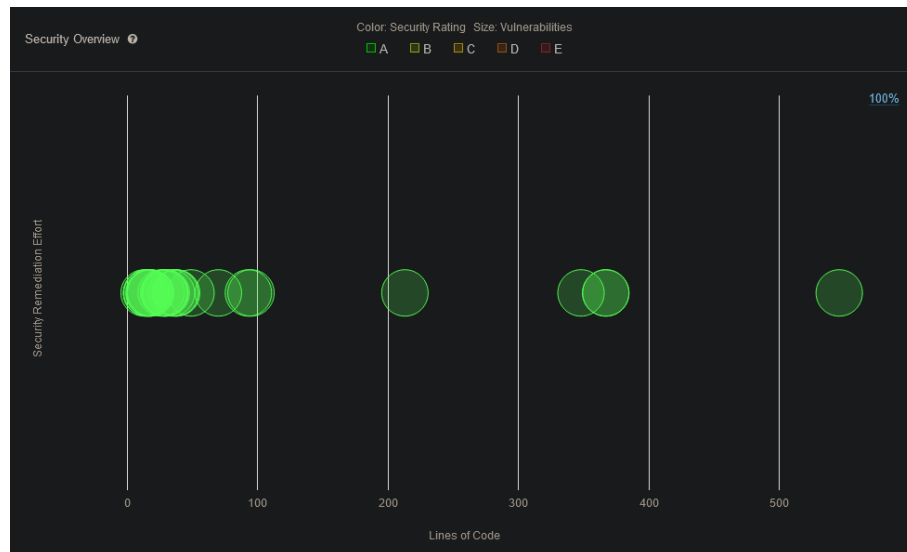
- Código duplicado



- Mantenibilidad



■ Seguridad



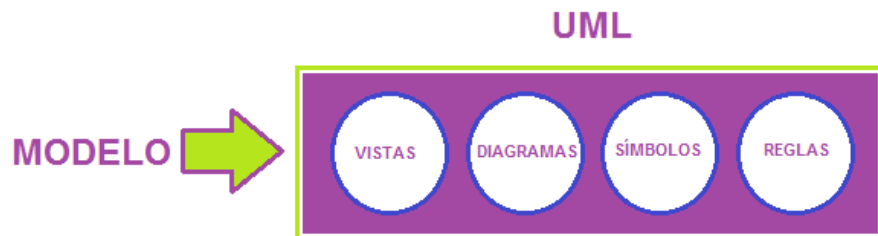
8.1. *Tecnología de información*

- **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.

8.2. *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.






9. Cronograma











9.1. Iteraciones

Iterations	Start Date	End Date
▼ ProyectoSistemaDiagnosticoMedico		
Iteration 1	02/11/2020	08/11/2020
Iteration 2	09/11/2020	15/11/2020
Iteration 3	... 16/11/2020	22/11/2020



















9.2. Tablero de iteraciones y duración

ProyectoSistemaDiagnosticoMedico Team Backlog		
Iteration 1 Planned Effort: -  3	Current	02/11/2020 - 08/11/2020 5 working days
Iteration 2 Planned Effort: -  7		09/11/2020 - 15/11/2020 5 working days
Iteration 3 Planned Effort: -  6		16/11/2020 - 22/11/2020 5 working days

















9.2.1. Iteración 1

 	Order	Title	State	Assigned To
	1	 Registrar empleados	  Active	Abraham LIPA CALABILLA
	2	 Registrar pacientes	 Active	Abraham LIPA CALABILLA
	3	 Registrar usuarios	 Active	Abraham LIPA CALABILLA

9.2.2. Iteración 2

 	Order	Title	State	Assigned To
	1	 Registrar diagnóstico de ...	  New	KATERIN ALMENDRA MERINO QUISPE
	2	 Acceder mediante un Login	 New	EDWART JUAN BALCÓN COAHILA
	3	 Listar pacientes	 New	Percy Huber TAQUILA CARAZAS
	4	 Listar usuarios	 New	Percy Huber TAQUILA CARAZAS
	5	 Listar empleados	 Active	Percy Huber TAQUILA CARAZAS
	6	 Listar diagnóstico	 New	KATERIN ALMENDRA MERINO QUISPE
	7	 Eliminar datos de diagnóstico	 New	KATERIN ALMENDRA MERINO QUISPE

9.2.3. Iteración 3

 	Order	Title	State	Assigned To
	1	 Actualizar datos de pacie...	  New	EDWART JUAN BALCÓN COAHILA
	2	 Actualizar datos de usuario	 New	EDWART JUAN BALCÓN COAHILA
	3	 Actualizar datos de empleado	 New	Abraham LIPA CALABILLA
	4	 Buscar empleado	 New	Abraham LIPA CALABILLA
	5	 Buscar usuario	 New	Lisbeth Isabel ESPINOZA CASO
	6	 Buscar paciente	 New	Lisbeth Isabel ESPINOZA CASO

9.2.4. Backlog

	Order	Work Item Type	Title	State	Story ...	Value Area
+	1	User Story	Registar diagnóstico de paciente	New		Business
	2	User Story	Acceder mediante un Login	New		Business
	3	User Story	Registrar empleados	Active		Business
	4	User Story	Listar pacientes	New		Business
	5	User Story	Actualizar datos de paciente	New		Business
	6	User Story	Registrar pacientes	Active		Business
	7	User Story	Listar usuarios	New		Business
	8	User Story	Actualizar datos de usuario	New		Business
	9	User Story	Registrar usuarios	Active		Business
	10	User Story	Listar empleados	Active		Business
	11	User Story	Listar diagnóstico	New		Business
	12	User Story	Eliminar datos de diagnóstico	New		Business
	13	User Story	Actualizar datos de empleado	New		Business
	14	User Story	Buscar empleado	New		Business
	15	User Story	Buscar usuario	New		Business
	16	User Story	Buscar paciente	New		Business

9.3. Tablero de historias

ProjectoSistemaDiagnosticoMedico Te...

☆ ⚙

Board

Analytics

View as Backlog

Stories

New

Active4/5

Resolved0/5

Closed

New Item

8 Registrar diagnóstico de paciente

StateNew

7 Acceder mediante un Login

StateNew

6 Listar pacientes

Percy Huber TAQUILA CARAZAS

StateNew

11 Actualizar datos de paciente

StateNew

5 Listar usuarios

Percy Huber TAQUILA CARAZAS

StateNew

10 Actualizar datos de usuario

StateNew

19 Listar diagnóstico

3 Registrar empleados

Abraham LIPA CALABILLA

StateActive

2 Registrar pacientes

Abraham LIPA CALABILLA

StateActive

1 Registrar usuarios

Abraham LIPA CALABILLA

StateActive

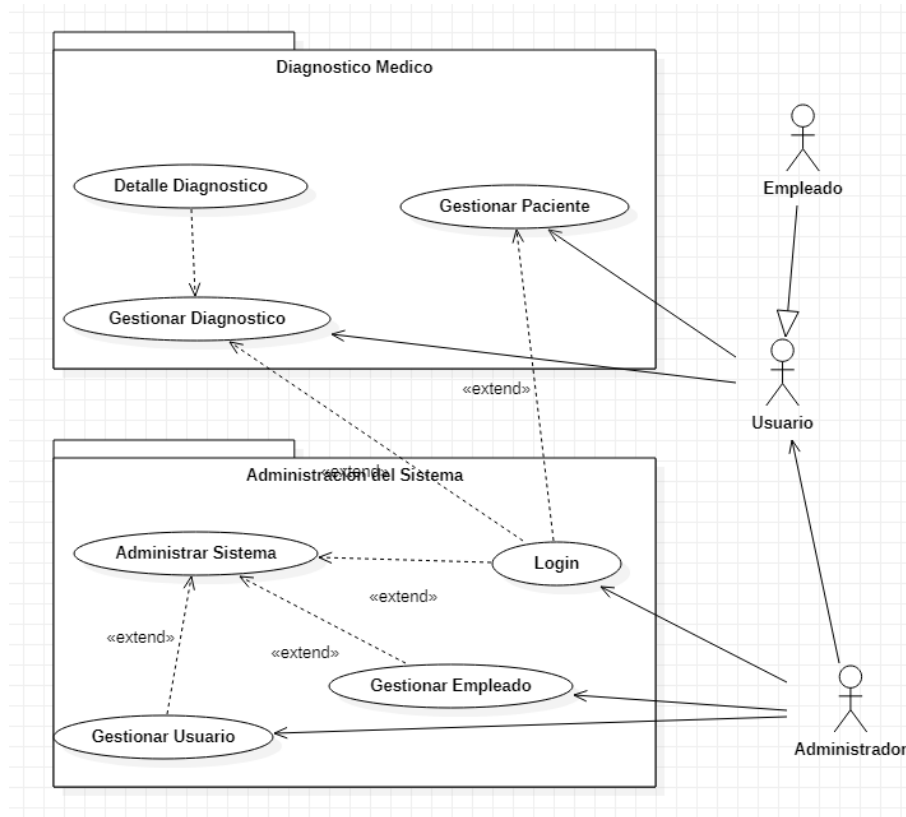
4 Listar empleados

Percy Huber TAQUILA CARAZAS

StateActive

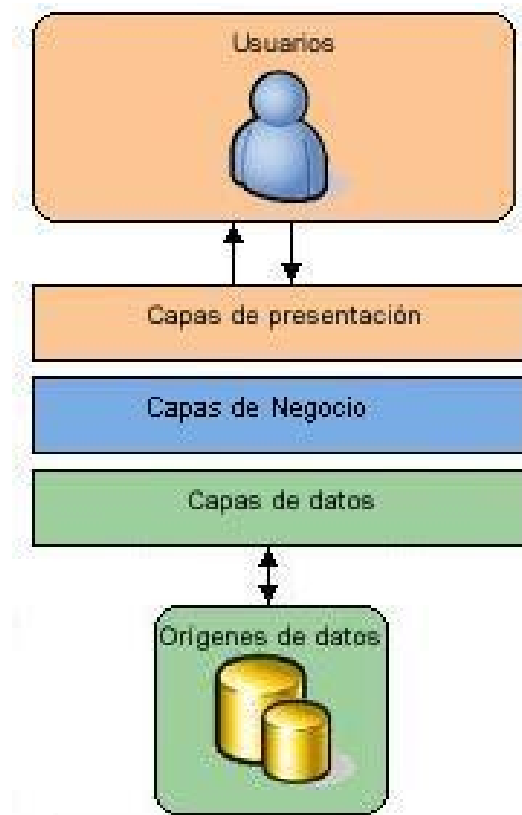
10. Desarrollo de Solución de Mejora

10.1. Casos de Uso de la aplicación

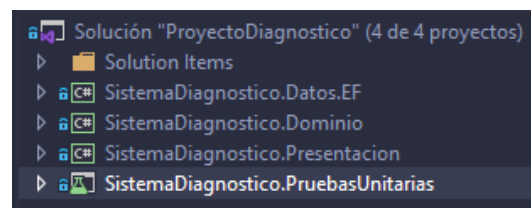


10.2. Diagrama de Arquitectura de la aplicación

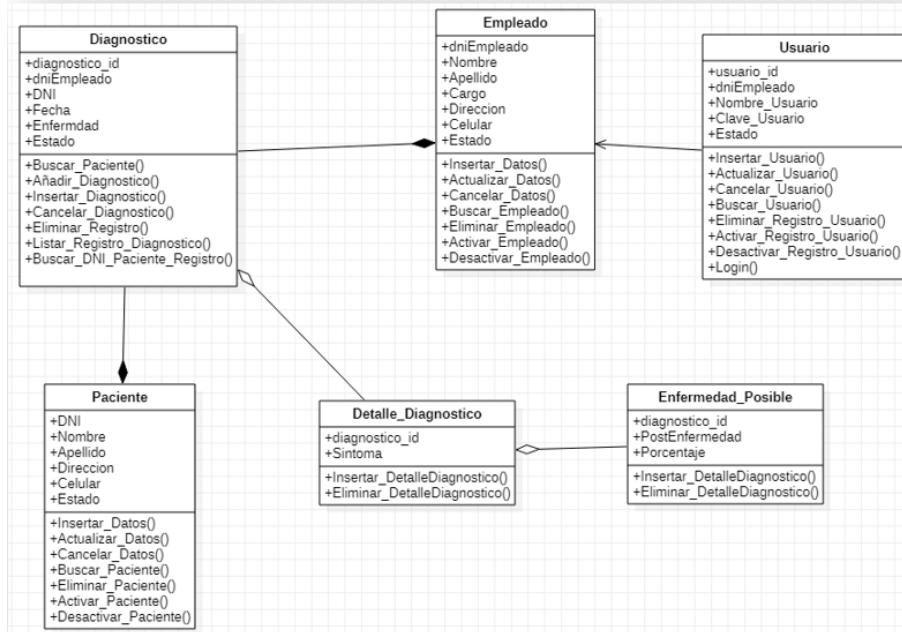
- Existen diferentes tipos de arquitectura o patrones a seguir para desarrollar un software, en este caso voy a explicar en que consiste la arquitectura de 3 Capas , que ami parecer es la mas general o la mas basica para desarrollar.
- La Capa de Presentación: Donde se encuentran los formularios y la 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