

UNIVERSIDAD PRIVADA-DE-TACNA



INGENIERIA DE SISTEMAS

TITULO:

RandyStore

CURSO:

BASE DE DATOS II

DOCENTE:

Ing. Patrick Cuadros Quiroga

Integrantes:

Mejía Rodriguez, Julio Oliver	(2010037899)
Paredes Catacora, Randi Angel	(2013047246)
Herrera Amezquita, Derian Francisco	(2017059489)

Índice

1. Introduccion	1
2. Titulo	1
3. Autores	1
4. Planteamiento del problema	1
4.1. Problema	1
4.2. Justificacion	1
4.3. Alcance	1
5. Objetivos	1
5.1. General	1
5.2. Especifico	1
6. Referencias Teoricas	1
7. Desarrollo de la prueba	1
7.1. Tecnologia de informacion	1
7.2. Metodología, tecnicas usadas	2
8. Diccionario de datos	9
9. Cadena de conexion	11

Resumen

El presente proyecto , desarrollo de una sistema de control de clientes, personal e inventario del gimnasio Randys busca contribuir automatizar los procesos del area de ventas de productos deportivos. Tambien busca llevar un control de los clientes y de el personal con el que cuenta la tienda.

Abstrac

The present project, development of a control system for clients, personnel and Randys gym inventory seeks to help automate processes in the area sales of sports products. It also seeks to keep track of customers and staff with whom the company has store.

1. Introduccion

2. Titulo

Sistema de control de inventario y personal RandyStore.

3. Autores

4. Planteamiento del problema

4.1. Problema

4.2. Justificacion

4.3. Alcance

5. Objetivos

5.1. General

5.2. Especifico

6. Referencias Teoricas

7. Desarrollo de la prueba

Para realizar las pruebas en nuestro codigo utilizaremos la herramienta de **SonarQube**.

7.1. Tecnologia de informacion

Utilizamos **SonarQube** que es una plataforma para evaluar codigo fuente.Es de uso libre y usa diversas herramientas de analisis estatico como Checkstyle, PMD o FindBugs para obtener métricas que pueden ayudar a mejorar la calidad del código de un programa.

Tambien usamos el repositorio de **Azure DevOps** para realizar la programacion de nuestro codigo fuente en equipo.

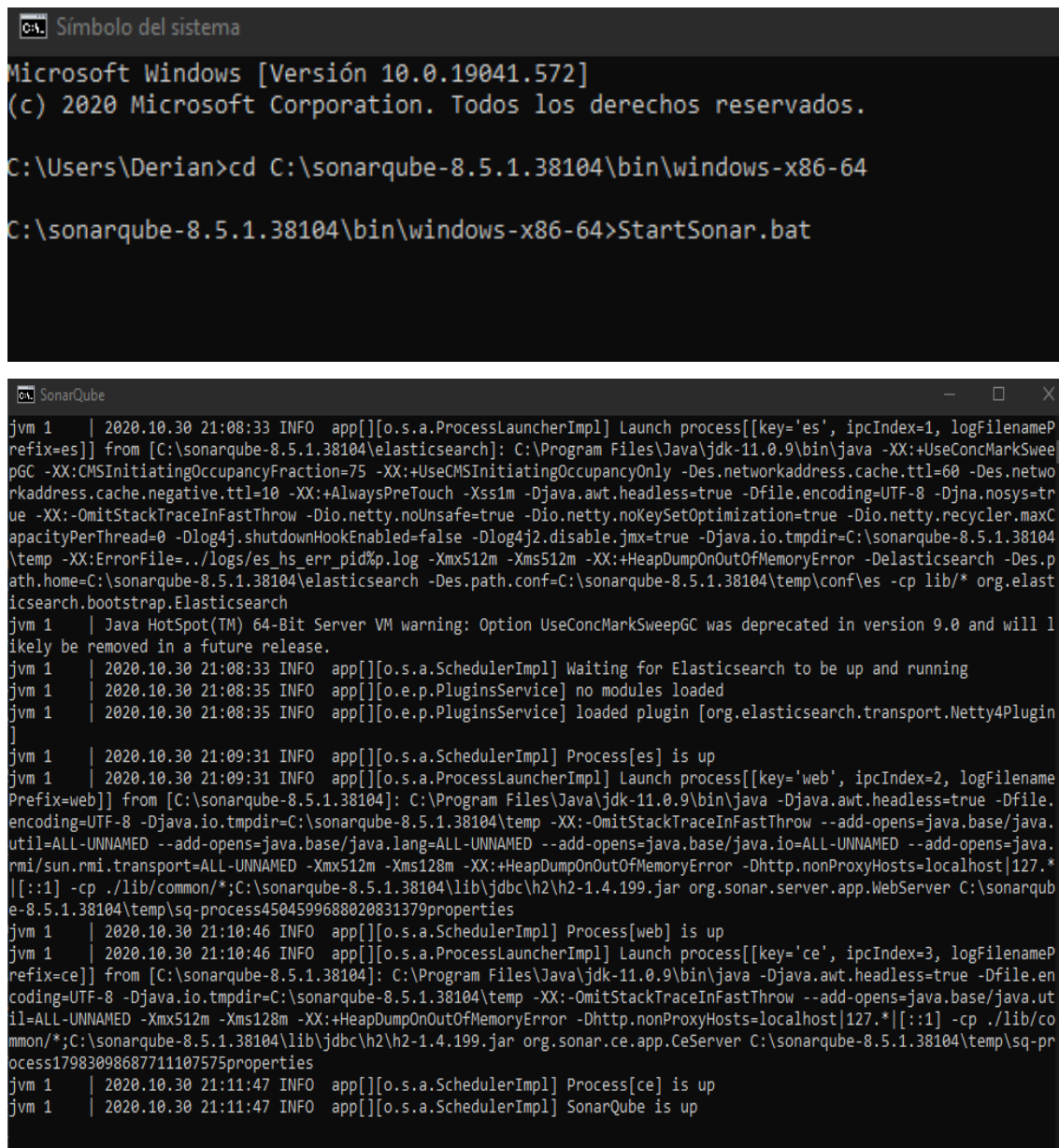
Usamos el compilador de codigo fuente de **Visual Studio 2017** para realizar la programacion del sistema.

El gestor de base de datos **SQL Server** para la creacion de la base de datos.

7.2. Metodología, tecnicas usadas

Proceso de evaluacion del codigo

- Primero descargamos **SonarQube** de manera local en nuestro equipo, luego nos dirigimos a la direccion donde guardamos la descarga en el cmd y ejecutamos el bat "StarSonar.bat" para iniciar el servicio.



```
C:\Users\Derian>cd C:\sonarqube-8.5.1.38104\bin\windows-x86-64
C:\sonarqube-8.5.1.38104\bin\windows-x86-64>StartSonar.bat
```

```
jvm 1 | 2020.10.30 21:08:33 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[[key='es', ipcIndex=1, logFilenamePrefix=es]] from [C:\sonarqube-8.5.1.38104\elasticsearch]: C:\Program Files\Java\jdk-11.0.9\bin\java -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djna.nosys=true -XX:-OmitStackTraceInFastThrow -Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=true -Dio.netty.recycler.maxCapacityPerThread=0 -Dlog4j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -Djava.io.tmpdir=C:\sonarqube-8.5.1.38104\temp -XX:ErrorFile=../logs/es_hs_err_pid%p.log -Xmx512m -Xms512m -XX:+HeapDumpOnOutOfMemoryError -Delasticsearch -Des.path.home=C:\sonarqube-8.5.1.38104\elasticsearch -Des.path.conf=C:\sonarqube-8.5.1.38104\temp\conf\es -cp lib/* org.elasticsearch.bootstrap.Elasticsearch
jvm 1 | Java HotSpot(TM) 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
jvm 1 | 2020.10.30 21:08:33 INFO app[[o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
jvm 1 | 2020.10.30 21:08:35 INFO app[[o.e.p.PluginsService] no modules loaded
jvm 1 | 2020.10.30 21:08:35 INFO app[[o.e.p.PluginsService] loaded plugin [org.elasticsearch.transport.Netty4Plugin]
jvm 1 | 2020.10.30 21:09:31 INFO app[[o.s.a.SchedulerImpl] Process[es] is up
jvm 1 | 2020.10.30 21:09:31 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[[key='web', ipcIndex=2, logFilenamePrefix=web]] from [C:\sonarqube-8.5.1.38104]: C:\Program Files\Java\jdk-11.0.9\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\sonarqube-8.5.1.38104\temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|:::1 -cp ./lib/common/*;C:\sonarqube-8.5.1.38104\lib\jdbc\h2\h2-1.4.199.jar org.sonar.server.app.WebServer C:\sonarqube-8.5.1.38104\temp\squid-process450459688020831379properties
jvm 1 | 2020.10.30 21:10:46 INFO app[[o.s.a.SchedulerImpl] Process[web] is up
jvm 1 | 2020.10.30 21:10:46 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[[key='ce', ipcIndex=3, logFilenamePrefix=ce]] from [C:\sonarqube-8.5.1.38104]: C:\Program Files\Java\jdk-11.0.9\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\sonarqube-8.5.1.38104\temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|:::1 -cp ./lib/common/*;C:\sonarqube-8.5.1.38104\lib\jdbc\h2\h2-1.4.199.jar org.sonar.ce.app.CeServer C:\sonarqube-8.5.1.38104\temp\squid-process17983098687711107575properties
jvm 1 | 2020.10.30 21:11:47 INFO app[[o.s.a.SchedulerImpl] Process[ce] is up
jvm 1 | 2020.10.30 21:11:47 INFO app[[o.s.a.SchedulerImpl] SonarQube is up
```

-Ahora analizaremos el código. Ingresamos a la dirección de nuestra solución y ejecutamos el código con el nombre del proyecto y el nombre de la solución.

```
C:\Users\Derian\Desktop\WTF\GIMNASIO_PROJECT\Proyecto_Gimnasio>SonarScanner.MSBuild.exe begin /k:"Proyecyo_Gimnasio" /n:"Proyecto_Gimnasio" /v:"2.0"
SonarScanner for MSBuild 4.10
Using the .NET Framework version of the Scanner for MSBuild
Pre-processing started.
Preparing working directories...
21:25:13.553 Updating build integration targets...
21:25:13.668 Fetching analysis configuration settings...
21:25:29.207 Provisioning analyzer assemblies for cs...
21:25:29.216 Installing required Roslyn analyzers...
21:25:30.426 Provisioning analyzer assemblies for vbnet...
21:25:30.436 Installing required Roslyn analyzers...
21:25:30.685 Pre-processing succeeded.
```

-Después ejecutamos el siguiente código para realizar el análisis

```
C:\Users\Derian\Desktop\WTF\GIMNASIO_PROJECT\Proyecto_Gimnasio>"C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\MSBuild\15.0\Bin\MSBuilder.exe" /t:Rebu...
```

-Y por último este código para finalizar el análisis.

```
C:\Users\Derian\Desktop\WTF\GIMNASIO_PROJECT\Proyecto_Gimnasio>SonarScanner.MSBuild.exe end
```

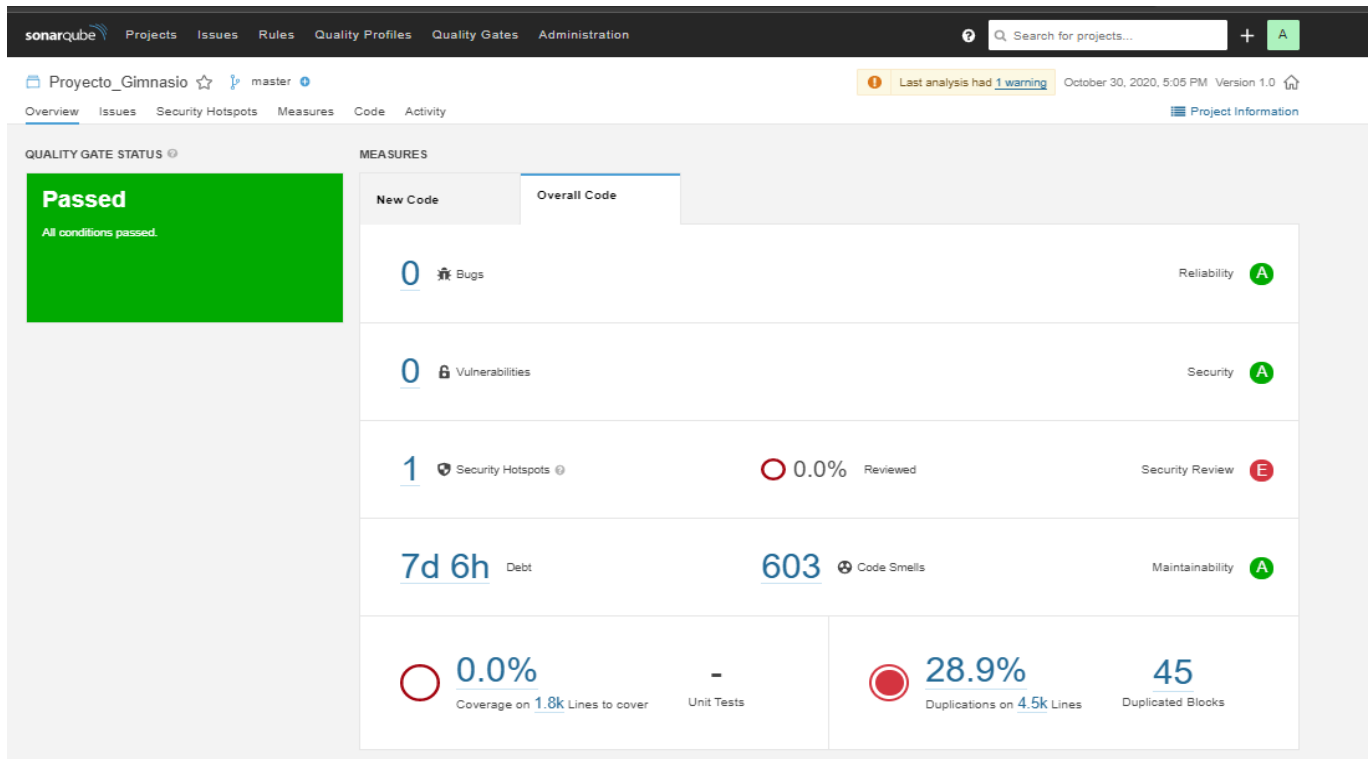
-Luego entramos a nuestro localhost de SonarQube. Aquí tenemos el análisis de nuestro proyecto anterior y el que mejoramos.

The screenshot shows the SonarQube web interface at localhost:9000/projects. The interface includes a navigation bar with tabs for Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. A search bar is present on the right. The main content area displays a list of projects. Two projects are visible:

- GimnasioPro**: Status **Passed**, Last analysis: 5 hours ago. Metrics: Bugs (37, C), Vulnerabilities (0, A), Hotspots Reviewed (1, A), Code Smells (77, A), Coverage (0.0%, R), Duplications (1.5%, G), Lines (9.5k, S). A note indicates the main branch is not analyzed yet, with a [Configure analysis](#) button.
- Proyecto_Gimnasio**: Status **Passed**, Last analysis: 4 hours ago. Metrics: Bugs (0, A), Vulnerabilities (0, A), Hotspots Reviewed (0.0%, E), Code Smells (603, A), Coverage (0.0%, R), Duplications (28.9%, R), Lines (4.5k, S).

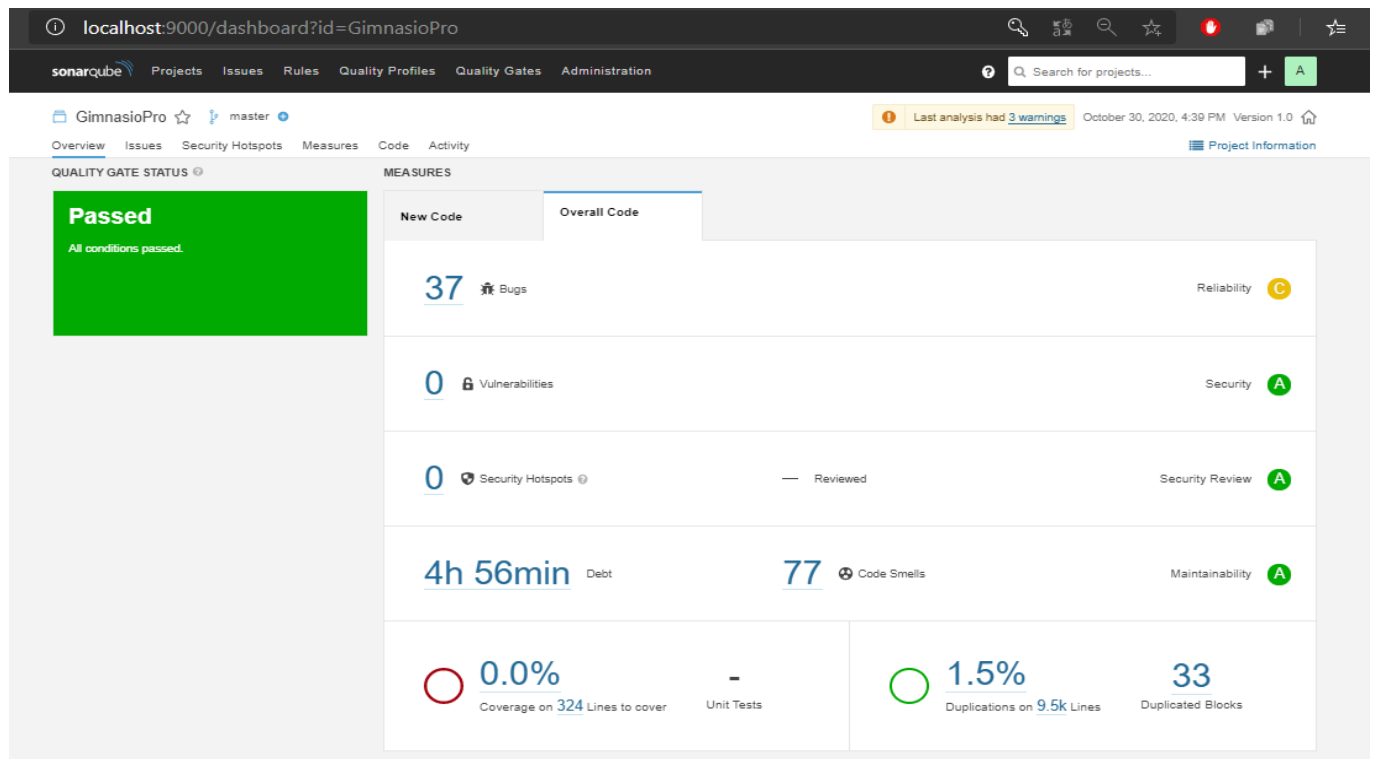
On the left, there are filters for Quality Gate (Passed: 2, Failed: 0), Reliability (A: 1, B: 0, C: 1, D: 0, E: 0), Security (A: 2, B: 0, C: 0, D: 0, E: 0), and Security Review (A: 1, B: 0).

-Muestra del analisis de nuestro proyecto anterior:

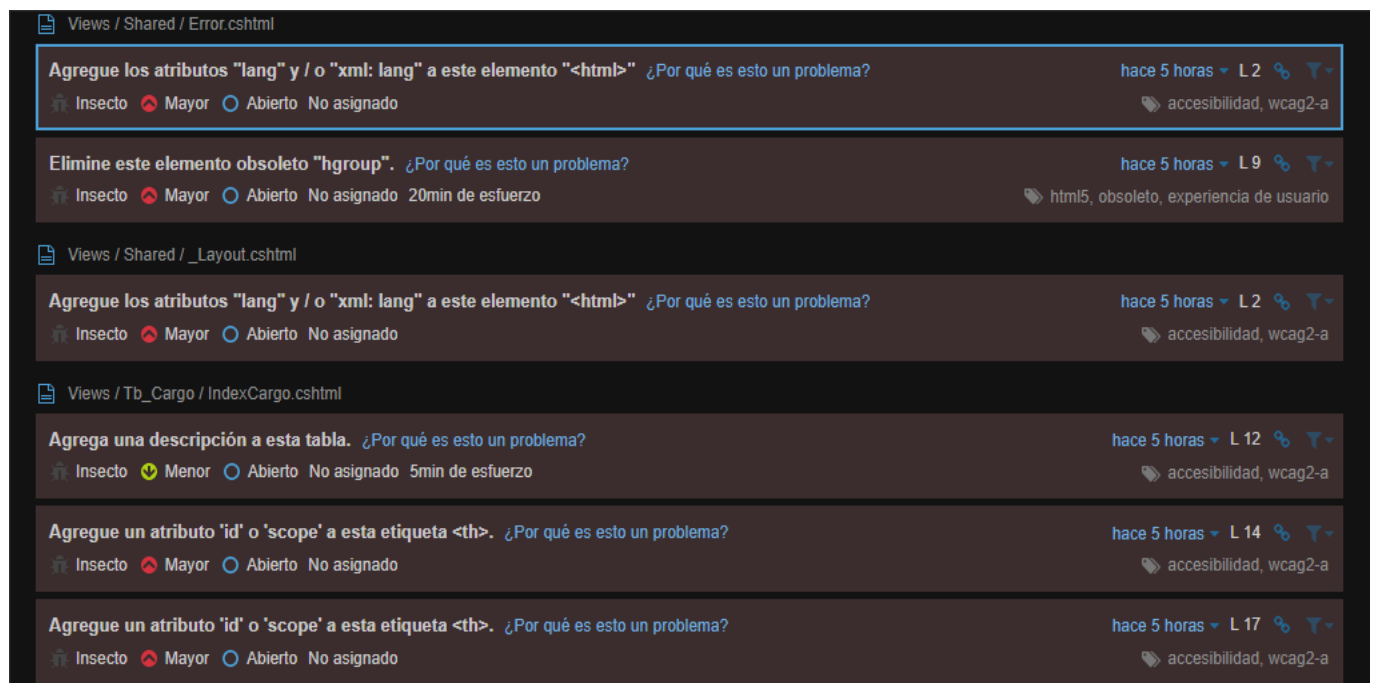


Como vemos tenemos 0 bugs, 0 vulnerabilidades, 1 punto de acceso de seguridad vulnerabilidad, Una deuda tecnica de 7 dias 6 horas , 603 de codigo que podria ser un problema, casi un 30 por-ciento de duplicidad de codigo en 45 bloques.

-Ahora veremos el código que realizamos con algunas mejoras como el uso de EntityFramework




En este caso nuestra solución tiene 37 bugs. Mas que todos en las vistas del proyecto. en donde nos pide poner un id y scope a nuestras tablas




En la parte de seguridad se eliminaron las vulnerabilidades.

0  Vulnerabilities

Security 

0  Security Hotspots 

— Reviewed

Security Review 

En lo que se refiere a deuda total tenemos 4h 56 min de deuda. Relacionados a 41 componentes del proyecto.

Technical Debt 4h 56min	
Controllers/Tb_ProveedorController.cs	45min
Controllers/Tb_ClienteController.cs	30min
Controllers/Tb_DetalleVentaController.cs	30min
Controllers/Tb_IngresoController.cs	30min
Controllers/Tb_ProductoController.cs	30min
Controllers/Tb_VentaController.cs	30min
App_Start/RouteConfig.cs	18min
Controllers/Tb_CargoController.cs	15min
Controllers/Tb_EmpleadoController.cs	15min
Controllers/Tb_TipoProductoController.cs	15min
App_Start/BundleConfig.cs	12min
App_Start/FilterConfig.cs	12min
Controllers/HomeController.cs	8min
Global.asax.cs	4min
Properties/AssemblyInfo.cs	2min
There are 41 hidden components with a score of 0. Show Them	

Se encontro tenemos 324 líneas por cubrir con pruebas unitarias estas relacionados con 14 componentes de nuestro proyecto.

Project Overview

> Reliability ?

> Security ?

> Security Review ?

> Maintainability ?

> Coverage **LINES TO COVER**

Overview

Overall

Coverage0.0%

Lines to Cover324

Uncovered Lines324

Line Coverage0.0%

> Duplications

> Size

> Complexity ?

> Issues

GimnasioPro

View as

List

↑ ↓ to select files

← → to navigate

14 files

Lines to Cover 324

Controllers/Tb_DetalleVentaController.cs

46

Controllers/Tb_ProductoController.cs

46

Controllers/Tb_VentaController.cs

46

Controllers/Tb_IngresoController.cs

42

Controllers/Tb_ProveedorController.cs

37

Controllers/Tb_EmpleadoController.cs

29

Controllers/Tb_ClienteController.cs

24

Controllers/Tb_CargoController.cs

18

Controllers/Tb_TipoProductoController.cs

18

App_Start/BundleConfig.cs

5

Controllers/HomeController.cs

5

Global.asax.cs

4

App_Start/RouteConfig.cs

3

App_Start/FilterConfig.cs

1

14 of 14 shown

Tenemos codigo peligro que puede dañar a un futuro el proyecto que tiene que ver en el uso innecesarios de codigo y agregar contrucctores de proteccion.

The screenshot displays the SonarQube interface for the 'GimnasioPro' project. The left sidebar shows the 'Issues' tab with filters for 'Type: CODE SMELL' (77 items) and 'Severity' (Minor: 58, Major: 19). The main panel lists several issues, including 'Remove this unnecessary "using"' and 'Rename class "Tb_CargoController" to match pascal case naming rules'. The issues are categorized by severity (Minor, Major) and status (Open, Not assigned). The interface also shows a 'Bulk Change' button and a 'Project Information' section.

Encontramos duplicidad de codigo producto de malas practicas al programar, pero esta en el rango de aceptacion



Medidas



8. Diccionario de datos

Tabla de cargos de empleados

Nombre de la Tabla:	Cargo	Fecha creacion: 18/11/2020
Descripción:	Tabla que contendra los tipos de cargos de empleado	
CAMPO	TIPO DE DATO	DESCRIPCION
IdCargo	int	Clave unica de cargo
NombreCargo	varchar	Nombre del cargo
Descripcion	varchar	Descripcion del cargo
Campos clave: IdCargo		

Tabla de categorias de producto

Nombre de la Tabla:	Categoria	Fecha creacion: 18/11/2020
Descripción:	Tabla que contendra los tipos de productos	
CAMPO	TIPO DE DATO	DESCRIPCION
IdCategoria	int	Clave unica de categoria
Descripcion	varchar	Descripcion de la categoria
Campos clave: IdCategoria		

Tabla de cliente

Nombre de la Tabla:	Cliente	Fecha creacion: 18/11/2020
Descripción:	Tabla que tendra los datos personales del cliente	
CAMPO	TIPO DE DATO	DESCRIPCION
IdCliente	int	Clave unica del cliente
Nombre	varchar	Nombre completo del cliente
Apaterno	varchar	Apellido paterno del cliente
Amaterno	varchar	Apellido materno del cliente
Direccion	varchar	Direccion del cliente
Dni	int	Numero de dni del cliente
Celular	int	Numero de telefono del cliente
Campos clave: IdEmpleado		

Tabla de comprobante

Nombre de la Tabla:	Comprobante	Fecha creacion: 18/11/2020
Descripción:	Tabla que contendra los comprobantes realizados	
CAMPO	TIPO DE DATO	DESCRIPCION
IdComprobante	int	Clave unica del comprobante
FechaEmision	DateTime	Fecha de creacion del comprobante
Total	decimal	Total de dinero a pagar
IdCliente	int	Clave referencial a la tabla cliente
IdEmpleado	int	Clave referencial a la tabla empleado
Campos clave:IdComprobante,IdEmpleado,IdCliente		

Tabla del detalle del comprobante

Nombre de la Tabla:	DetalleComprobante	Fecha creacion: 18/11/2020
Descripción:	Tabla que tendra el detalle de cada comprobante	
CAMPO	TIPO DE DATO	DESCRIPCION
IdDetalle	int	Clave unica de detalle
IdComprobante	int	Clave referencial a la tabla comprobante
IdProducto	int	Clave referencial a la tabla producto
Precio	decimal	Costo unitario del producto
Cantidad	int	Cantidad del producto vendido
Total	decimal	Total a pagar por los productos
Campos clave:IdComprobante,IdDetalle,IdProducto		

Tabla del empleado

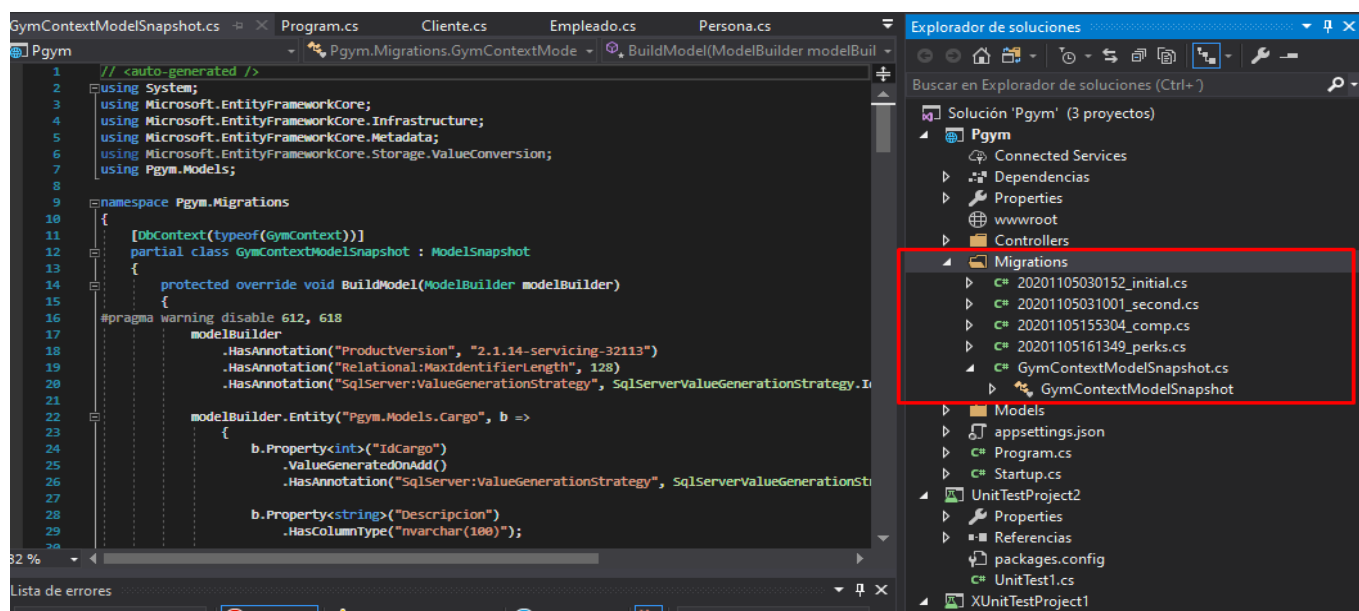
Nombre de la Tabla:	Empleado	Fecha creacion: 18/11/2020
Descripción:	Tabla que contendra datos personales del empleado	
CAMPO	TIPO DE DATO	DESCRIPCION
IdEmpleado	int	Clave unica del empleado
Nombre	varchar	Nombre completo del empleado
Apaterno	varchar	Apellido paterno del empleado
Amaterno	varchar	Apellido materno del empleado
Direccion	varchar	Direccion del empleado
Dni	int	Numero de dni del empleado
Celular	int	Numero de telefono del empleado
Clave	int	Clave generada para el empleado
IdCargo	int	Clave referencial a la tabla cargo
Campos clave: IdEmpleado, IdCargo		

Tabla del producto

Nombre de la Tabla:	Producto	Fecha creacion: 18/11/2020
Descripción:	Tabla que tendra informacion de los productos	
CAMPO	TIPO DE DATO	DESCRIPCION
IdProducto	int	Clave principal de producto
NombreProducto	varchar	Nombre del producto
Cantidad	varchar	Cantidad de stock del producto
PrecioVenta	varchar	Precio de venta del producto
IdCategoria	varchar	Clave referencial a la tabla Categoria
Campos clave: IdProducto,IdCategoria		

9. Cadena de conexion

DbContext donde esta el contexto de nuestra base de datos y las migraciones prueba que realizamos



Cadena de conexion para el orm que va en el json appsettings

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "Conexion": "Server=localhost; Database=Agym; Trusted_Connection=True; MultipleActiveResultSets=True;"
  }
}
```

Luego en la clase startup se coloca la cadena de conexion string que hicimos anteriormente para que el sistema lo reconozca y nuestro DbContext tambien

```
17 public class Startup
18 {
19     public Startup(IConfiguration configuration)
20     {
21         Configuration = configuration;
22     }
23
24     public IConfiguration Configuration { get; }
25
26     // This method gets called by the runtime. Use this method to add services to the container.
27     public void ConfigureServices(IServiceCollection services)
28     {
29         services.AddCors();
30         services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
31
32         //Se agrega esto
33         services.AddDbContext<GymContext>(options =>
34             options.UseSqlServer(Configuration.GetConnectionString("Conexion")));
35     }
36
37     // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
38     public void Configure(IApplicationBuilder app, IHostingEnvironment env)
39     {
40         app.UseCors(options =>
41         {
42             options.WithOrigins("http://localhost:3000");
43             options.AllowAnyMethod();
44             options.AllowAnyHeader();
45         });
46     }
47 }
```

Nuestras migraciones se realizaron para SQLServer