



Sistema de Administración de Propiedad Horizontal (Appartment)

Andrés Felipe Díaz Leal

José Alberto Mendoza Ocampo

Código

Universidad Antonio Nariño

Programa Especialización en Ingeniería de Software

Facultad de Sistemas

Bogotá, Colombia

2023

Sistema de Administración de Propiedad Horizontal

Andrés Felipe Díaz Leal

José Alberto Mendoza Ocampo

Proyecto de grado presentado como requisito parcial para optar al título de:
Especialista de Ingeniería de Software

Director (a):

Dianalin Neme Prada

Universidad Antonio Nariño

Programa Especialización en Ingeniería de Software

Facultad de Sistemas

Bogotá, Colombia

2023

NOTA DE ACEPTACIÓN

El trabajo de grado titulado Sistema de Administración
de Propiedad Horizontal,
Cumple con los requisitos para optar
El título de Especialista de Ingeniería de Software.

Firma del Tutor

Firma Jurado

Firma Jurado

Tabla de Contenido

Resumen.....	9
Abstract.....	10
Introducción	11
1. Sistema de Administración de Propiedad Horizontal	12
1.1 Formulación y descripción del problema	12
2. Objetivos	13
2.1 Objetivo general.....	13
2.1 Objetivos específicos.....	13
3. Marco de Referencia.....	14
3.1. Estado del Arte.....	14
3.2. Impacto	17
3.3. Componente de Innovación.....	18
3.4. Marco teórico	19
3.4.1. Marco legal	19
3.4.2. Metodología Scrum.....	20
3.4.3. Marco Técnico.....	22
3.4.3.1 Ingeniería de software	22
3.4.3.2 .NET	23
3.4.3.3 SQL server 2019	24
3.4.3.4 SonarQube	25
3.4.3.5 Azure DevOps.....	26
4. Proceso de software.....	27
4.1. Stakeholders	28
4.2. Requerimientos Funcionales	29
4.3. Requerimientos No Funcionales	45

4.4. Diseño y arquitectura	46
4.5. Diagrama de despliegue.....	47
4.6. Diagrama de secuencia	48
4.7. Diagrama de clases.....	49
4.8. Arquitectura de alto nivel	51
5. Construcción.....	52
6. Pruebas.....	54
6.1. Pruebas de proceso	54
6.1. Pruebas de producto y seguridad	57
7. Instalación y Configuración	59
8. Conclusiones	66
Referencias Bibliográficas.....	67

Lista de ilustraciones

<i>Ilustración 1. Login Fuente: Elaboración Propia</i>	30
<i>Ilustración 2. Recuperar contraseña. Fuente: Elaboración Propia</i>	30
<i>Ilustración 3. Correo recuperar contraseña. Fuente: Elaboración Propia</i>	30
<i>Ilustración 4. Cambio de contraseña. Fuente: Elaboración Propia</i>	31
<i>Ilustración 5. Home. Fuente: Elaboración Propia</i>	31
<i>Ilustración 6. Registro de persona - propietario. Fuente: Elaboración Propia</i>	32
<i>Ilustración 7. Listado de personas – propietario. Fuente: Elaboración Propia</i>	33
<i>Ilustración 8. Asociar apartamento a propietario. Fuente: Elaboración Propia</i>	33
<i>Ilustración 9. Listado de residentes. Fuente: Elaboración Propia</i>	33
<i>Ilustración 10. Registro de persona - arrendatario. Fuente: Elaboración Propia</i>	34
<i>Ilustración 11. Listado de personas – arrendatario. Fuente: Elaboración Propia</i>	35
<i>Ilustración 12. Asociar apartamento a arrendatario. Fuente: Elaboración Propia</i>	35
<i>Ilustración 13. Listado de residentes. Fuente: Elaboración Propia</i>	35
<i>Ilustración 14. Crear empleado. Fuente: Elaboración Propia</i>	36
<i>Ilustración 15. Empleados. Fuente: Elaboración Propia</i>	37
<i>Ilustración 16. Registro de visita. Fuente: Elaboración Propia</i>	38
<i>Ilustración 17. Registro de huésped. Fuente: Elaboración Propia</i>	39
<i>Ilustración 18. Listado de huéspedes. Fuente: Elaboración Propia</i>	39
<i>Ilustración 19. Eliminar registro huésped. Fuente: Elaboración Propia</i>	40
<i>Ilustración 20. Ingreso huésped. Fuente: Elaboración Propia</i>	40
<i>Ilustración 21. Registro de visitante. Fuente: Elaboración Propia</i>	41
<i>Ilustración 22. Listado de visitantes. Fuente: Elaboración Propia</i>	42
<i>Ilustración 23. Eliminar registro de visitante. Fuente: Elaboración Propia</i>	42
<i>Ilustración 24. Crear zona común. Fuente: Elaboración Propia</i>	43
<i>Ilustración 25. Listado de zonas. Fuente: Elaboración Propia</i>	43
<i>Ilustración 26. Crear reserva de zona común. Fuente: Elaboración Propia</i>	44
<i>Ilustración 27. Registro de zonas comunes. Fuente: Elaboración Propia</i>	44
<i>Ilustración 28. Diseño y arquitectura. Fuente: Elaboración Propia</i>	46
<i>Ilustración 29. Diagrama de despliegue. Fuente: Elaboración Propia</i>	47
<i>Ilustración 30. Diagrama de secuencia 1. Fuente: Elaboración Propia</i>	48
<i>Ilustración 31. Diagrama de secuencia 2. Fuente: Elaboración Propia</i>	49
<i>Ilustración 32. Diagrama de clases 1. Fuente: Elaboración Propia</i>	50
<i>Ilustración 33. Diagrama de clases 2. Fuente: Elaboración Propia</i>	50
<i>Ilustración 34. Arquitectura de alto nivel. Fuente: Elaboración Propia</i>	51
<i>Ilustración 35. Capas implementadas a nivel de código. Fuente: Elaboración Propia</i>	52
<i>Ilustración 36. Pruebas SonarQube 1. Fuente: Elaboración Propia</i>	54
<i>Ilustración 37. Pruebas SonarQube 2. Fuente: Elaboración Propia</i>	55
<i>Ilustración 38. Pruebas SonarQube 3. Fuente: Elaboración Propia</i>	55
<i>Ilustración 39. Pruebas SonarQube 4. Fuente: Elaboración Propia</i>	56

<i>Ilustración 40. Pruebas SonarQube 5. Fuente: Elaboración Propia</i>	<i>56</i>
<i>Ilustración 41. Local Host. Fuente: Elaboración Propia</i>	<i>57</i>
<i>Ilustración 42 Pruebas Owasp Zap. Fuente: Elaboración Propia</i>	<i>58</i>
<i>Ilustración 43 Instalación y configuración 1. Fuente: Elaboración Propia</i>	<i>59</i>
<i>Ilustración 44 Instalación y configuración 2. Fuente: Elaboración Propia</i>	<i>60</i>
<i>Ilustración 45 Instalación y configuración 3. Fuente: Elaboración Propia</i>	<i>60</i>
<i>Ilustración 46 Instalación y configuración 4. Fuente: Elaboración Propia</i>	<i>61</i>
<i>Ilustración 47 Instalación y configuración 5. Fuente: Elaboración Propia</i>	<i>62</i>
<i>Ilustración 48 Instalación y configuración 6. Fuente: Elaboración Propia</i>	<i>62</i>
<i>Ilustración 49 Instalación y configuración 7. Fuente: Elaboración Propia</i>	<i>63</i>
<i>Ilustración 50 Instalación y configuración 8. Fuente: Elaboración Propia</i>	<i>63</i>
<i>Ilustración 51 Instalación y configuración 9. Fuente: Elaboración Propia</i>	<i>64</i>
<i>Ilustración 52 Instalación y configuración 10. Fuente: Elaboración Propia</i>	<i>64</i>
<i>Ilustración 53 Instalación y configuración 11. Fuente: Elaboración Propia</i>	<i>65</i>

Lista de tablas

<i>Tabla 1. Estado del arte. Fuente: Elaboración propia.....</i>	15
<i>Tabla 2. Definición de ingeniería de software. Fuente: Elaboración propia</i>	22
<i>Tabla 3. Identificación de StakeHolders. Fuente: Elaboración propia.....</i>	28
<i>Tabla 4. HU: Ingreso al sistema. Fuente: Elaboración propia.....</i>	29
<i>Tabla 5. HU: Registro de propietarios. Fuente: Elaboración propia</i>	32
<i>Tabla 6. HU: Registro de Arrendatarios. Fuente: Elaboración propia</i>	34
<i>Tabla 7. HU: Registro de Vigilante. Fuente: Elaboración propia</i>	36
<i>Tabla 8. HU: Historial de visitas. Fuente: Elaboración propia.....</i>	37
<i>Tabla 9. HU: Registro de huésped. Fuente: Elaboración propia.....</i>	39
<i>Tabla 10. HU: Registro de visitante. Fuente: Elaboración propia</i>	41
<i>Tabla 11. HU: Reserva de Zonas Comunes. Fuente: Elaboración propia.....</i>	43

Resumen

En lugares como Bogotá cada día es más fácil encontrar una gran variedad de conjuntos y edificios que presentan inconvenientes al momento de llevar un seguimiento de las personas que visitan las instalaciones, dado a que dichos procesos se registran de manera manual y en muchas ocasiones se pierden los registros realizados por parte de los residentes y vigilantes.

Dichos problemas se incrementaron con la llegada de nuevas aplicaciones para renta de apartamentos, ya que los administradores se vieron en la necesidad de llevar un registro de las personas que ocuparían temporalmente dichas copropiedades, con el fin de evitar problemas de acceso y seguridad.

Con el fin de solventar este tipo de inconvenientes, este proyecto tiene la finalidad de ofrecer una aplicación web que ofrezca a los propietarios de los apartamentos hacer el registro de huéspedes que vienen de aplicaciones como Airbnb y Booking y así mismo otros residentes puedan realizar registros de visitas y zonas comunes, para que los administradores y vigilantes puedan llevar un seguimiento y control más seguro de las instalaciones.

Abstract

In places like Bogotá, every day it is easier to find a wide variety of complexes and buildings that present problems when keeping track of the people who visit the facilities, given that these processes are recorded manually and are often lost. the records carried out by residents and security guards.

These problems increased with the arrival of new applications for apartment rentals, since the administrators found it necessary to keep a record of the people who would temporarily occupy said co-properties, in order to avoid access and security problems.

In order to solve this type of problem, this project aims to offer a web application that allows apartment owners to register guests who come from applications such as Airbnb and Booking and also other residents can register guests. visits and common areas, so that administrators and security guards can monitor and control the facilities more safely.

Introducción

El presente proyecto tiene la finalidad de simplificar y mejorar significativamente procesos de control que hoy en día se realizan de manera manual en las copropiedades, ya que hoy en día una gran cantidad de propietarios se ven en la necesidad de reportar de manera eficiente la ocupación temporal de huéspedes provenientes de plataformas como Airbnb y Booking.

Actualmente, muchos conjuntos residenciales y edificios de interés social no cuentan con los recursos suficientes para solventar grandes plataformas para administración de propiedad horizontal y a corte de 2021 según el ministerio de vivienda, ciudades como Bogotá contaban con más de 209.638 viviendas nuevas vendidas de enero a noviembre de ese mismo año (*Con 209.638 viviendas nuevas vendidas en los últimos 11 meses, el 2021 ya es el mejor año en ventas para Colombia / MinVivienda, s. f.-b*), logrando identificar un gran mercado potencial que necesita implementar este tipo de aplicativos en sus instalaciones.

Por lo anterior, se pretende suministrar una Aplicación Web Responsive que permita a los administradores y usuarios de la copropiedad, optimizar sus procesos de control de visitantes y reserva de zonas comunes de una manera fácil y segura, en la que se brinde una experiencia de autogestión más satisfactoria.

1. Sistema de Administración de Propiedad Horizontal

1.1 Formulación y descripción del problema

El presente proyecto se realiza con la finalidad de facilitar procesos administrativos que se realizan por parte de la administración y habitantes de la copropiedad, puesto que en muchos casos existen procesos manuales y susceptibles a errores propios de la manualidad, que dilatan la atención y respuesta oportuna por parte de la administración a los interesados de la Copropiedad.

Un caso de ellos nace a partir del surgimiento de aplicaciones para ocupación de vivienda temporal como Airbnb o Booking, donde los propietarios y apoderados de los apartamentos se ven en la necesidad de anunciar a la administración y vigilantes, quienes van a estar ocupando su propiedad durante cierta cantidad de tiempo.

Por lo anteriormente mencionado, se requiere automatizar los procesos realizados actualmente por el administrador como, control e ingreso de visitantes, asignación y reserva de zonas comunes.

Teniendo en cuenta los inconvenientes anteriormente mencionados, ¿Como mejorar el ingreso de visitantes en las propiedades horizontales de una manera más eficaz, eficiente y segura?

2. Objetivos

2.1 Objetivo general

Desarrollar una Aplicación Web Responsive que permita a los administradores y usuarios de la copropiedad, la gestión de sus procesos por medio de un canal digital donde se lleve un control de visitantes y reserva de zonas comunes de una manera fácil y segura.

2.1 Objetivos específicos

- Proveer una plataforma tecnológica que permita, autogestión por parte de los propietarios y/o residentes en el registro de ingreso de visitantes a la copropiedad por medio de un canal virtual.
- Ofrecer a propietarios y/o residentes opciones de consulta y reserva de las Zonas comunes.
- Disponer de una aplicación web que permita el acceso a cualquiera de los residentes de una copropiedad.
- Gestionar los permisos de acceso y disponibilidad de la información de la copropiedad.
- Proveer una interfaz de cara a los vigilantes, que facilite la validación y revisión de la información de los visitantes y reservas de Zonas comunes.

3. Marco de Referencia

3.1. Estado del Arte

Para la ejecución de este proyecto, se realizó un análisis general de algunas de las plataformas que más se utilizan actualmente en el mercado para la administración de conjuntos en propiedad horizontal y algunas de sus principales características. Lo cual suministro información valiosa que permite conocer las necesidades con que cuentan los administradores y propiedades horizontales, y las carencias en las plataformas existentes y de qué manera poder suministrar una herramienta con un factor adicional de Innovación y muy competitivo en el mercado.

Nombre	Funcionalidades	Enlace
<p data-bbox="305 1136 459 1205">Edificios Davivienda</p> 	<ul data-bbox="509 1136 1081 1604" style="list-style-type: none">● Manejo de la cartera.● Conciliación automática de los recaudos.● Publicación de comunicados.● Mensajes privados.● Pago de la administración con débito y crédito.● Reserva de zonas comunes.	<p data-bbox="1130 1136 1438 1205">https://www.edificiosdavivienda.com</p>

<p>Edifito</p> 	<ul style="list-style-type: none"> ● Registro de Proveedores. ● Noticias y Asambleas. ● Informes de Gestión Administrativa. ● Reserva de zonas comunes. ● Alerta de emergencias. 	<p>https://edifito.co</p>
<p>Propiedata</p> 	<ul style="list-style-type: none"> ● Reservación de zonas comunes. ● Solicitudes al administrador (PQR). ● Novedades del conjunto. ● Votaciones en línea. ● Control de visitantes. ● Registro de mascotas. ● Biblioteca de documentos. 	<p>https://www.propiedata.com</p>
<p>Metro cuadrado</p> 	<ul style="list-style-type: none"> ● Publicación de comunicados. ● Registro de visitantes con QR. ● Reserva de zonas comunes. ● Chat con la administración. ● Recepción de paquetes. ● Simulador de gastos. 	<p>https://www.metrocuadrado.com</p>

Tabla 1. Estado del arte. Fuente: Elaboración propia

De acuerdo con la investigación realizada, se encuentra que la mayor parte de aplicaciones que existen actualmente en el mercado, están enfocadas en facilitar el pago de la administración, publicar comunicados y hacer reserva de las zonas comunes.

En cuanto a control de visitantes, son muy pocas las aplicaciones que se especializan en ofrecer dicho servicio, motivo por el cual se encuentra una oportunidad de intervenir en dicho segmento para captar la atención de conjuntos que presentan inconvenientes para control de visitantes.

3.2. Impacto

Actualmente muchos conjuntos de la ciudad de Bogotá están acoplados a un modelo de ingreso en el que los vigilantes por medio de un citófono deben anunciar la llegada de pedidos o visitantes, pero en muchas ocasiones los residentes se encuentran ocupados o fuera de las instalaciones, lo cual ocasiona problemas de ingreso en la recepción.

Con la creación de la herramienta web, se pretende intervenir en el control de registro de las copropiedades de una manera fácil y efectiva, agilizando el ingreso de visitantes y la reserva de zonas comunes para los siguientes interesados:

Administradores:

- Registro de propietarios.
- Consulta de visitas
- Autorización y control de uso en zonas comunes

Residentes:

- Agilidad en el ingreso de visitantes.
- Registro de huéspedes temporales.
- Separación de zonas comunes.

Vigilantes:

- Registro y control de visitantes.
- Procesos de ingresos más rápidos.

3.3. Componente de Innovación

A raíz del surgimiento de aplicaciones como Booking y Airbnb, los administradores de los conjuntos se vieron en la necesidad de exigir a los anfitriones los datos de las personas que van a ingresar a las instalaciones para evitar el ingreso de personas no deseadas.

Actualmente no existe una plataforma que dentro de sus funcionalidades implemente el registro de usuarios que vengan de estas plataformas y de esta manera evitar que se hagan filas o se diligencien formularios largos en papel para autorizar el ingreso.

Es por esto por lo que, al ofrecer una aplicación fácil e intuitiva a los propietarios y administradores de los predios para el registro de visitantes, otorga una experiencia de uso en la que se ahorran tiempos, se aumenta la seguridad de la copropiedad y se lleva un mejor control de las personas que se pueden llegar a hospedar.

3.4. Marco teórico

3.4.1. Marco legal

El presente proyecto busca generar una herramienta que permita gestionar y dar cumplimiento a las medidas entregadas por la ley 675 de 2001, la cual define o establece la normativa que aplica sobre los bienes o inmuebles que se conciben como régimen de propiedad horizontal.

Adicionalmente todo el manejo de información y protección de esta se fundamenta sobre la ley 1581 del 2012 Correspondiente a la ley de protección de Datos personales de Colombia, dentro de la cual se establece las normas y principios, para brindar un manejo adecuado de la información personal de todos los ciudadanos y otorgar el derecho a la privacidad y la protección de datos personales.

Dentro del control de ingreso y salida de visitantes y personas en general, se tiene como base o referencia legal la Ley 675 de 2001. Art. 63.03 de agosto de 2001 (Colombia). por medio de la cual se definen y establecen los mecanismos de control apropiados o acordes a los establecido por la norma y velando por el cumplimiento de la reglamentación o normativa interna de la copropiedad.

3.4.2. Metodología Scrum

Se propone como marco de trabajo metodológico el uso de Scrum, teniendo en cuenta que permite realizar la organización de las tareas y gestión de los requerimientos dentro un marco de agilidad y sobre el cual se tiene como base que son equipos autogestionados y comprometidos, trabajando colaborativamente en el objetivo de un fin y que en cada iteración o Sprint, se genera un incremento de producto funcional que nos permite generar valor o ganancias tempranas y que son percibidas por nuestros interesados.

Los Sprints, se ejecutan en ciclos cortos y de duración fija de manera iterativa, y los mismos tendrán una duración de 3 semanas. Cada iteración completada debe entregar un incremento de producto o producto mínimo viable, con el menor esfuerzo al cliente cuando lo solicite.

El proceso parte de la lista de requisitos priorizada, en esta lista el Product Owner establece una prioridad a los requisitos basados en el valor que le aportan respecto a su costo o esfuerzo y quedan repartidos en sprints y entregas.

Se realiza una reunión de planeación o planificación del Sprint donde se presenta al equipo el listado de requisitos que previamente fueron priorizados por el Product Owner y en esta misma se aclaran las dudas y se definen los requisitos que se encontrarán incluidos dentro del mismo y que podrán ser entregados al cliente como MVP.

Una vez identificados los requisitos, se procede a realizar la definición de las tareas que serán trabajadas dentro de la duración del sprint por parte del Team Developer. Estas tareas una vez iniciado el sprint se realizan reuniones diarias de seguimiento (Dailys) con el fin de poder tener de primera mano el avance y conocer los posibles impedimentos o bloqueantes sobre la ejecución de las tareas, los cuales serán gestionados por el facilitador

o Scrum Master del equipo, con el fin de mantener el foco del equipo en el desarrollo de sus actividades y facilitando la adopción y conocimiento de la metodología.

Durante la ejecución del Sprint en conjunto con el Product Owner y el Team, se refinan los requisitos de los próximos Sprints, con el fin de garantizar y maximizar la utilidad y funcionalidad de lo que se desarrolla y que se obtenga una ganancia o incremento que genere valor.

El último día del Sprint se realiza una reunión con el fin de presentar al Product Owner el incremento del producto a liberar y validando que se cumpliera el objetivo planeado, además de revisar si es necesario reevaluar o replantear algún tipo de acción para el próximo sprint.

Finalmente se realiza por parte del Scrum Master y el Team Developer una reunión de retrospectiva en donde se analiza cómo se ha estado trabajado y los que posibles problemas u oportunidades para tener en cuenta en la ejecución del próximo sprint, con el fin de ir mejorando en cada una de las iteraciones que se van generando y adicionalmente identificar y escalar los posibles obstáculos que puedan afectar la productividad del equipo en la consecución de los objetivos.

- Scrum Máster: Jose Alberto Mendoza Ocampo
- Team Developers: Andrés Felipe Díaz, José Alberto Mendoza Ocampo

3.4.3. Marco Técnico

3.4.3.1 Ingeniería de software

Autor	Definición	Referencia
Sommerville, I. (2005). Ingeniería del software. Pearson educación.	La ingeniería de software es una dependencia de la ingeniería que controla varios aspectos en la ejecución de software, desde las etapas iniciales procura utilizar la especificación del sistema hasta el mantenimiento posterior su debido uso.	Link
Martínez, R. N., Fuentes, J. R. L., Durango, A., & Ramos, D. (2015). Curso de Ingeniería de Software. IT Campus Academy.	La ingeniería de software es una forma de ingeniería que utiliza herramientas de cómputo y las matemáticas para resolver diferentes problemas de software con una mejor relación en cuanto a costo-beneficio. Por lo tanto, se trata de la aplicación correctiva de software para el desarrollo, operación y mantenimiento.	Link

Tabla 2. Definición de ingeniería de software. Fuente: Elaboración propia

3.4.3.2 .NET

Se trata de una plataforma de desarrollo de software que fue creada por Microsoft con el objetivo de unir diferentes variedades de catálogos de productos, dentro de sus funcionalidades incluye herramientas de desarrollo y varios sistemas operativos para optimizar procesos. La creación de .NET tenía como objetivo permitir la creación de aplicaciones y sistemas que fueran independientes a la arquitectura física y del sistema o en el que se encuentra desarrollando.

Se pueden realizar diferentes desarrollos web e incluso programas dedicados a escritorio o aplicaciones para dispositivos móviles. Además, fomenta el desarrollo en gran variedad de plataformas, como permitir la comunicación entre apps en diferentes sistemas operativos móviles, dentro de los que encontramos IOS o Android. Algunos de sus beneficios son:

- La posibilidad de usar las varias funcionalidades ya diseñadas permite a los diseñadores realizar proyectos de importación y adaptación a las necesidades en lugar de crear desde cero.
- Acortar el tiempo de ejecución de los proyectos.
- Facilitar la gestión de mantenimiento de las aplicaciones creadas en esta plataforma.
- Disminución de costos como resultado de la reducción de los tiempos de desarrollo y mantenimiento.

Recuperado el 21 de octubre de 2021, de <https://www.emagister.com/blog/que-es-el-net-para-que-sirve/>

3.4.3.3 SQL server 2019

Se trata de un RDBMS, o sistema de gestión de bases de datos relacionales, que puede usarse en entornos informáticos de corporaciones para el procesamiento de diferentes transacciones, la gestión empresarial y el análisis.

Microsoft SQL Server se basa en SQL, al igual que otros programas RDBMS, pero se basa en manejar un lenguaje de programación estándar que los administradores de bases de datos y otros profesionales de tecnologías de la información principalmente utilizan para administrar bases de datos y consultar sus datos.

De igual manera, por medio de Big Data otorga a los usuarios unir contenedores de SQL Server, Spark y HDFS. Además, incorpora el enmascaramiento de datos estáticos y la reconstrucción de índices. De igual manera con la recuperación de datos también realiza y elimina una fase de rehacer en el número de secuencias de registros de página más antiguos. Esto se hace, cuando un usuario cierra una aplicación que ha estado funcionando durante mucho tiempo, para que no tenga que esperar mucho tiempo a que esta tenga que ser cerrada para su uso.

Recuperado el 13 de abril de 2023, de

<https://www.computerweekly.com/es/definicion/Microsoft-SQL-Server>

3.4.3.4 SonarQube

Se trata de una herramienta que proporciona retroalimentación del código mediante un proceso de análisis estáticos en el que se identifican diferentes aspectos sensibles en los que se puede llegar a mejorar. Esta tarea ayudará a adquirir gran variedad de métricas que permiten la optimización del código.

SonarQube tiene muchas características que hacen más fácil su correcto funcionamiento y otorga grandes ventajas al usuario, como que es ideal para la integración continua. Básicamente, se dedica a compilar y ejecutar pruebas en el proyecto que vamos a realizar, con el objetivo de detectar errores, fallos o vulnerabilidades del sistema antes de que nuestro aplicativo salga a producción.

Al tratarse de una herramienta enfocada en la corrección del código de forma automatizada, se ofrece la posibilidad de integrarla dentro de nuestro flujo de trabajo, lo que ayudará en las labores de análisis e inspección continua del código nuestro proyecto, así como en las solicitudes relacionadas con la inclusión de modificaciones.

Recuperado el 22 de julio de 2022, de <https://keepcoding.io/blog/que-es-sonarqube/>

3.4.3.5 Azure DevOps

La plataforma de desarrollo de software manejada en la nube Azure DevOps ofrece unas alternativas integrales para planificar, desarrollar, entregar y supervisar el ciclo de vida del software. Es una herramienta que permitirá a nuestro equipo de desarrollo de software trabajar juntos para desarrollar y entregar aplicaciones mucho más rápido y eficiente.

Esta plataforma crea un entorno flexible y colaborativo que ayuda a nuestro proyecto dentro del ciclo de vida de desarrollo. Se integra con la mayoría de las herramientas de desarrollo más populares en el mercado, admite Git e integración continua y ofrece herramientas Agile para la planificación, comunicación y seguimiento del trabajo para alcanzar los objetivos definidos de manera mucho más efectiva.

Por medio de esta herramienta nuestro equipo podrá mejorar el rendimiento y producir un producto de mejor calidad en menos tiempo, siempre y cuando se adopten las culturas, prácticas y herramientas DevOps

Recuperado el 08 de agosto de 2022, de <https://sentry.io/blog/que-es-azure-devops/>

4. Proceso de software

A continuación, se procederá a realizar la explicación del proceso de desarrollo de la aplicación, junto con los respectivos ítems que se abordaran para el correcto funcionamiento del software, dentro de los que se encuentran:

- **Análisis:** Identificación de interesados y levantamiento de requerimientos transformados en historias de usuario.
- **Diseño:** Opciones de arquitectura a abordar y explicación general de la estructura del software.
- **Construcción:** Selección de herramientas, aplicativos, lenguajes de programación.
- **Pruebas:** Validación de la calidad y seguridad del software.
- **Instalación o despliegue:** Explicación de cómo poner el software en funcionamiento.

4.1. Stakeholders

Interesado	Compromiso					Poder/Influencia	Interés
	Desconoce	Se resiste	Neutral	Apoya	Líder		
Administradores				X	D	A	A
Arrendatarios		X		D		B	B
Propietarios			X	D		A	A
Huéspedes	X			D		A	A
Visitantes	X			D		B	B
Vigilantes			X		D	A	A

Tabla 3. Identificación de Stakeholders. Fuente: Elaboración propia

Notas:

X: Actual; D: Deseado

A: Alto; B: Bajo

Dentro de la matriz de interesados se logra identificar que los usuarios más interesados en la adquisición de este tipo de aplicación estarían entre los administradores, propietarios, vigilantes y huéspedes que estarían ocupando los apartamentos, dado a que son los perfiles más afectados en procesos manuales al momento de gestionar reservas en zonas comunes y reportar visitas de personas que vienen de aplicaciones para renta de apartamentos, por lo cual se toman como eje para el registro de historias de usuarios.

4.2. Requerimientos Funcionales

Historias de Usuario			
No	1	Usuario:	Administrador
Título	Ingreso al sistema		
Características/Funcionalidad			
Yo como usuario administrador, requiero ingresar al sistema para poder hacer uso de las funcionalidades de la aplicación de manera segura.			
Razón/resultados			
Para llevar el control de las personas que ingresan al sistema.			
Criterios de Aceptación			
Funcional	X		No Funcional
No	Título	Contexto	Evento
1	Ingreso exitoso	Cuando el ingreso del usuario y contraseña son correctos.	El sistema permitirá el ingreso al sistema.
2	Ingreso fallido	Cuando el ingreso del usuario y contraseña son incorrectos.	El sistema NO permitirá el ingreso y el sistema presentará una alerta con el siguiente mensaje: "Usuario y/o contraseña no válida"
3	Tiempo de sesión.	Cuando el usuario supere más de una hora se debe cerrar la sesión.	El sistema NO permitirá el ingreso y devolverá al usuario al login.
4	Cambio de contraseña	Cuando el usuario ingrese por primera vez la contraseña provisional.	El sistema debe exigir el cambio por una nueva.
5	Recuperación de contraseña	Cuando el usuario no recuerde su contraseña.	Podrá solicitar él envió de una contraseña provisional al correo electrónico personal y debe ser cambiada una vez la ingrese.

Tabla 4. HU: Ingreso al sistema. Fuente: Elaboración propia

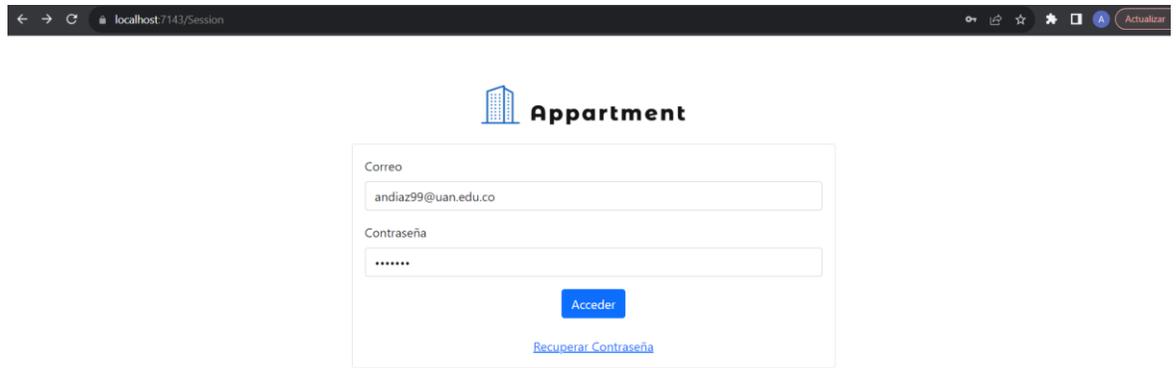


Ilustración 1. Login Fuente: Elaboración Propia

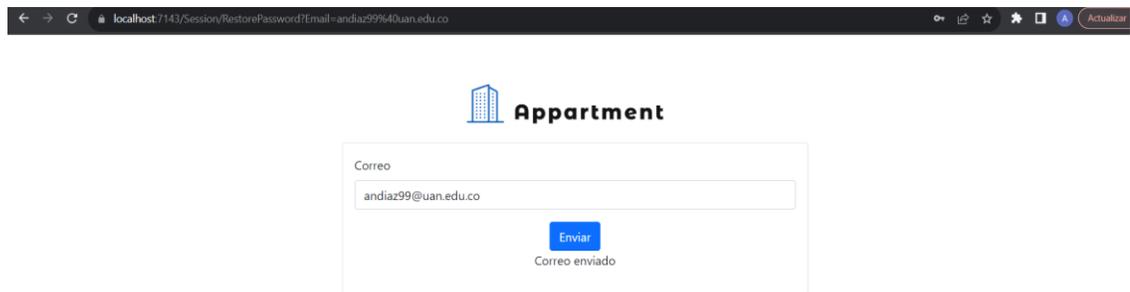


Ilustración 2. Recuperar contraseña. Fuente: Elaboración Propia

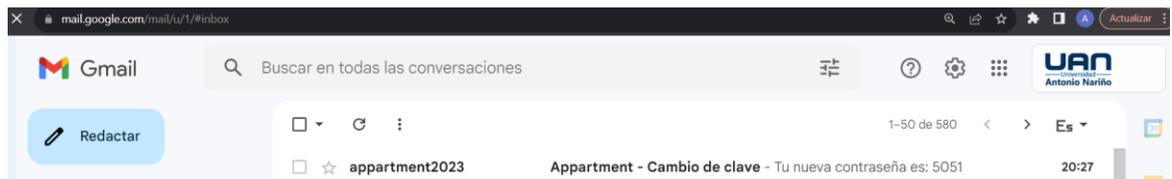


Ilustración 3. Correo recuperar contraseña. Fuente: Elaboración Propia

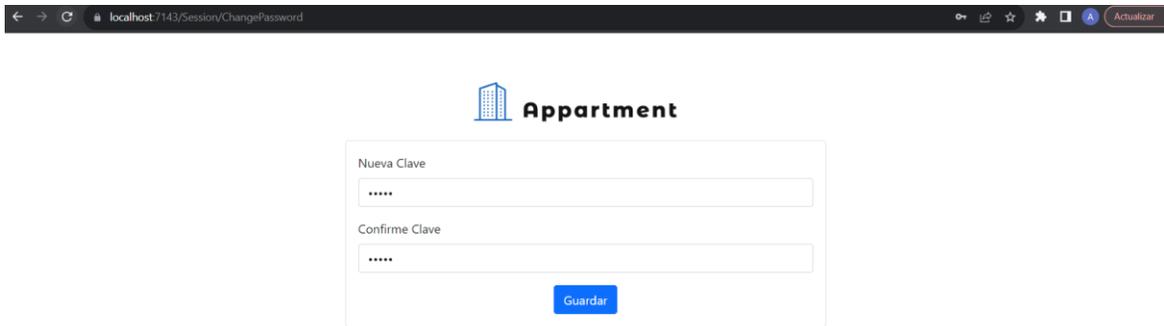


Ilustración 4. Cambio de contraseña. Fuente: Elaboración Propia



Ilustración 5. Home. Fuente: Elaboración Propia

Historias de Usuario			
No	2	Usuario:	Administrador
Título	Registro de propietarios		
Características/Funcionalidad			
Yo como usuario administrador, requiero registrar los datos de propietarios para poder identificar y actualizar los datos en el sistema.			
Razón/resultados			
Almacenar y editar los datos de los propietarios de la copropiedad			
Criterios de Aceptación			
Funcional	X		No Funcional
No	Título	Contexto	Evento
1	Almacenamiento de información.	Registrar la información de un nuevo propietario.	El sistema deberá solicitar los siguientes datos. <ul style="list-style-type: none"> • Nombres • Apellidos • Tipo de Documento

			<ul style="list-style-type: none"> • Numero de Documento • Teléfono • email • Fecha de Nacimiento • Tipo de Persona • Número Inmueble
2	Editar información	Corregir o actualizar datos de los propietarios almacenados en el sistema.	Se deberá incluir un botón de editar en el que el administrador pueda cambiar los datos almacenados.
3	Matriz de información.	Visualizar los datos registrados.	El sistema deberá incluir un listado en el que se visualicen los propietarios registrados.

Tabla 5. HU: Registro de propietarios. Fuente: Elaboración propia

The screenshot shows a web browser window with the URL localhost:7143/People/Edit/1. The page title is 'Appartment Inicio'. The main content is a form titled 'Persona' with the following fields and values:

- Nombre: ANDRÉS FELIPE
- Apellidos: DÍAZ LEAL
- Tipo de Documento: CC
- Número de Documento: 123456789
- Fecha de Nacimiento: 12/02/2023 06:03 p. m.
- Correo: andiaz99@uan.edu.co
- Teléfono: 113456789
- Activo:

At the bottom of the form, there are two buttons: 'Editar' (green) and 'Volver' (blue). The footer of the page reads '© 2023 - Appartment.Web - Privacy'.

Ilustración 6. Registro de persona - propietario. Fuente: Elaboración Propia

Nombre	Apellidos	Tipo Documento	Número Documento	Fecha de Nacimiento	Correo	Teléfono	Fecha Creación	Ultima Actualización	Estado
ANDRÉS FELIPE	DÍAZ LEAL	CC	123456789	12/02/2023 6:03:00 p. m.	andiaz99@uan.edu.co	113456789	12/09/2023 6:33:23 p. m.	26/10/2023 8:06:53 p. m.	<input checked="" type="checkbox"/>
JULIO	MENDIETA	CC	1231123	13/06/2023 9:18:00 p. m.	julio1234@gmail.com	3212456789	24/09/2023 9:18:46 p. m.	6/11/2023 4:22:27 p. m.	<input checked="" type="checkbox"/>
JOSE	MENDOZA	CC	72021795	29/04/1983 12:00:00 a. m.	jomendoza95@uan.edu.co	3103405345	5/11/2023 5:24:21 p. m.	6/11/2023 4:21:56 p. m.	<input checked="" type="checkbox"/>

Ilustración 7. Listado de personas – propietario. Fuente: Elaboración Propia

Ilustración 8. Asociar apartamento a propietario. Fuente: Elaboración Propia

Tipo	Activo	Apartamento	Persona
ARRENDATARIO	<input checked="" type="checkbox"/>	TORRE UNO - A Apto 101	JULIO MENDIETA
PROPIETARIO	<input checked="" type="checkbox"/>	TORRE UNO - A Apto 101	ANDRÉS FELIPE DÍAZ LEAL

Ilustración 9. Listado de residentes. Fuente: Elaboración Propia

Historias de Usuario			
No	3	Usuario:	Administrador
Título	Registro de Arrendatarios		
Características/Funcionalidad			
Yo como usuario administrador, requiero registrar los datos de arrendatarios para poder identificar y actualizar los datos en el sistema.			

Razón/resultados			
Almacenar y editar los datos de los arrendatarios de la copropiedad			
Criterios de Aceptación			
Funcional	X		No Funcional
No	Título	Contexto	Evento
1	Almacenamiento de información.	Registrar la información de un nuevo arrendatario.	<p>El sistema deberá solicitar los siguientes datos.</p> <ul style="list-style-type: none"> • Nombres • Apellidos • Tipo de Documento • Numero de Documento • Teléfono • email • Tratamiento • Fecha de Nacimiento • Tipo de Persona • Número Inmueble
2	Editar información	Corregir o actualizar datos de los arrendatarios almacenados en el sistema.	Se deberá incluir un botón de editar en el que el administrador pueda cambiar los datos almacenados.
3	Matriz de información.	Visualizar los datos registrados.	El sistema deberá incluir un listado en el que se visualicen los propietarios registrados.

Tabla 6. HU: Registro de Arrendatarios. Fuente: Elaboración propia

The screenshot shows a web browser window with the URL localhost:7143/People/Edit/2. The page title is 'Appartment' with navigation links for 'Inicio' and 'Salir'. The main content is a form titled 'Persona' with the following fields:

- Nombre: JULIO
- Apellidos: MENDIETA
- Tipo de Documento: CC
- Número de Documento: 1231123
- Fecha de Nacimiento: 13/06/2023 09:18 p. m.
- Correo: julio1234@gmail.com
- Teléfono: 3212456789
- Activo

At the bottom of the form are two buttons: 'Editar' (green) and 'Volver' (blue). The footer contains the text '© 2023 - Appartment.Web - Privacy'.

Ilustración 10. Registro de persona - arrendatario. Fuente: Elaboración Propia

Nombre	Apellidos	Tipo Documento	Número Documento	Fecha de Nacimiento	Correo	Teléfono	Fecha Creación	Última Actualización	Estado
ANDRÉS FELIPE	DÍAZ LEAL	CC	123456789	12/02/2023 6:03:00 p. m.	andiaz99@uan.edu.co	113456789	12/09/2023 6:33:23 p. m.	26/10/2023 8:06:53 p. m.	<input checked="" type="checkbox"/> Editar
JULIO	MENDIETA	CC	1231123	13/06/2023 9:18:00 p. m.	julio1234@gmail.com	3212456789	24/09/2023 9:18:46 p. m.	6/11/2023 4:22:27 p. m.	<input checked="" type="checkbox"/> Editar
JOSE	MENDOZA	CC	72021795	29/04/1983 12:00:00 a. m.	jomendoza95@uan.edu.co	3103405345	5/11/2023 5:24:21 p. m.	6/11/2023 4:21:56 p. m.	<input checked="" type="checkbox"/> Editar

Ilustración 11. Listado de personas – arrendatario. Fuente: Elaboración Propia

Ilustración 12. Asociar apartamento a arrendatario. Fuente: Elaboración Propia

Tipo	Activo	Apartamento	Persona	
ARRENDATARIO	<input checked="" type="checkbox"/>	TORRE UNO - A Apto 101	JULIO MENDIETA	Eliminar
PROPIETARIO	<input checked="" type="checkbox"/>	TORRE UNO - A Apto 101	ANDRÉS FELIPE DÍAZ LEAL	Eliminar

Ilustración 13. Listado de residentes. Fuente: Elaboración Propia

Historias de Usuario			
No	4	Usuario:	Administrador
Título	Registro de Vigilante		
Características/Funcionalidad			

Yo como usuario administrador, requiero registrar los datos de vigilante para poder identificar y actualizar los datos en el sistema.

Razón/resultados			
Almacenar y editar los datos de los vigilantes de la copropiedad			
Criterios de Aceptación			
Funcional	X		No Funcional
No	Título	Contexto	Evento
1	Almacenamiento de información.	Registrar la información de un nuevo vigilante.	<p>El sistema deberá solicitar los siguientes datos.</p> <ul style="list-style-type: none"> • Nombres • Apellidos • Tipo de Documento • Numero de Documento • Teléfono • email • Fecha de Nacimiento • Tipo de Persona
2	Editar información	Corregir o actualizar datos de los vigilantes almacenados en el sistema.	Se deberá incluir un botón de editar en el que el administrador pueda cambiar los datos almacenados.
3	Matriz de información.	Visualizar y filtrar los datos registrados.	El sistema deberá incluir una opción de filtros para que la consulta sea más fácil.

Tabla 7. HU: Registro de Vigilante. Fuente: Elaboración propia

Ilustración 14. Crear empleado. Fuente: Elaboración Propia

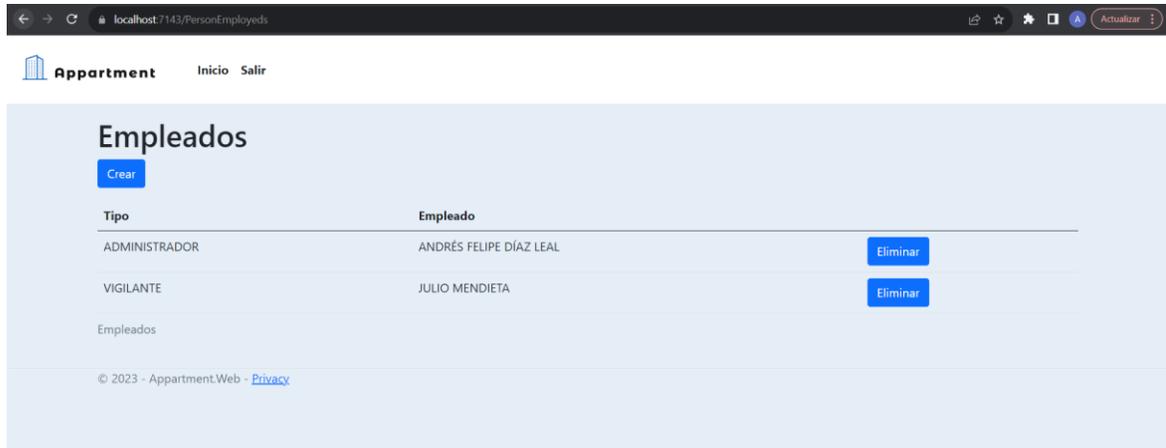


Ilustración 15. Empleados. Fuente: Elaboración Propia

Historias de Usuario			
No	5	Usuario:	Administrador o vigilante
Título	Historial de visitas		
Características/Funcionalidad			
Yo como usuario administrador o vigilante, deseo poder acceder a un historial de todas las visitas registradas Para llevar un control de ingreso de las personas que van a ingresar a las instalaciones			
Razón/resultados			
Control y seguimiento de visitas			
Criterios de Aceptación			
Funcional	X		No Funcional
No	Título	Contexto	Evento
1	Visualización de visitas.	Visualizar los registros de visitas realizadas por los usuarios	Mostrar un calendario con la relación de todas las visitas registradas, ordenadas por fecha y hora.
3	Editar información	Corregir o actualizar datos de las visitas almacenadas en el sistema.	Se deberá incluir un botón de editar en el que el usuario pueda cambiar los datos almacenados.
4	Confirmación de llegada.	El vigilante podrá dar aviso de la llegada del visitante en el sistema	Implementar un botón de confirmación al momento de la llegada del visitante.

Tabla 8. HU: Historial de visitas. Fuente: Elaboración propia



Ilustración 16. Registro de visita. Fuente: Elaboración Propia

Historias de Usuario			
No	6	Usuario:	Propietario
Título	Registro de huésped		
Características/Funcionalidad			
Yo como usuario propietario, requiero poder registrar la información básica de un huésped que va a ocupar temporalmente mi apartamento por medio de aplicaciones como Airbnb o Booking			
Razón/resultados			
Para que la seguridad del edificio esté informada de dichas visitas y permita el ingreso			
Criterios de Aceptación			
Funcional	X		No Funcional
No	Título	Contexto	Evento
1	Almacenamiento de información.	Registrar la información de un huésped.	<p>El sistema deberá solicitar los siguientes datos.</p> <ul style="list-style-type: none"> • Nombres • Apellidos • Tipo de Documento • Numero de Documento • Teléfono • Fecha de ingreso • Fecha de salida • Hora de ingreso • Hora de salida • Origen de Reserva • Código de reserva • Número Inmueble

2	Listado de registros	Visualizar en una lista los registros de huésped realizados.	Mostrar una vista de cara al propietario en el que sea fácil visualizar los registros de huéspedes.
3	Código de registro.	Al finalizar el formulario se debe contar con un código que debe ser suministrado por parte de los huéspedes al vigilante.	Generar un código único de reserva para brindar mayor seguridad al momento de ingreso.
4	Eliminar registro	El usuario puede tener acceso a eliminar el registro, en caso de que el huésped ya no pueda asistir.	Incluir un botón de eliminar en el listado de registros.

Tabla 9. HU: Registro de huésped. Fuente: Elaboración propia

Registrar Huesped

Nombre: MANUEL, Apellido: CORREA, Tipo Doc: CC, Número: 1233412466, Apartamento: TORRE UNO - A 101, Desde: 13/11/2023 03:00 p. m., Hasta: 16/11/2023 12:00 a. m., Origen: AIRBNB, Telefono: 3244567567

Buttons: Create, Volver

© 2023 - Appartment.Web - [Privacy](#)

Ilustración 17. Registro de huésped. Fuente: Elaboración Propia

Huespedes

Crear

Nombre y Apellido	Tipo Doc	Número	Desde	Hasta	Reserva No.	Ingresó	Apartamento	Origen
MANUEL CORREA	CC	1233412466	13/11/2023 12:00:00 a. m.	16/11/2023 12:00:00 a. m.	AIR-2-171230	<input type="checkbox"/>	1A 101	AIRBNB

Buttons: Editar, Eliminar

© 2023 - Appartment.Web - [Privacy](#)

Ilustración 18. Listado de huéspedes. Fuente: Elaboración Propia

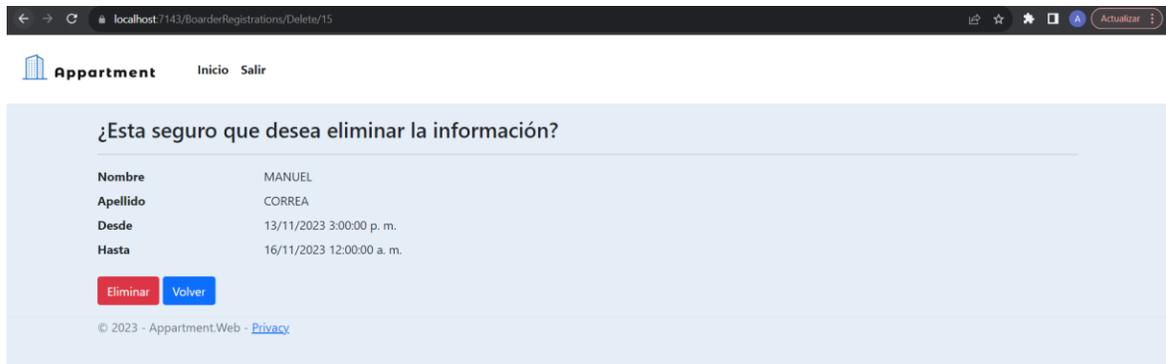


Ilustración 19. Eliminar registro huésped. Fuente: Elaboración Propia



Ilustración 20. Ingreso huésped. Fuente: Elaboración Propia

Historias de Usuario			
No	7	Usuario:	Propietario o arrendatario
Título	Registro de visitante		
Características/Funcionalidad			
Yo como usuario propietario, requiero poder registrar la información básica de un visitante que va a ingresar al apartamento.			
Para que la seguridad del edificio esté informada de dichas visitas y permita el ingreso			
Criterios de Aceptación			
Funcional	X		No Funcional
No	Título	Contexto	Evento
1	Almacenamiento de información.	Registrar la información de un visitante.	El sistema deberá solicitar los siguientes datos. <ul style="list-style-type: none"> • Nombres • Apellidos • Tipo de Documento

			<ul style="list-style-type: none"> • Numero de Documento • Teléfono (opcional) • Tratamiento • Foto (opcional) • Fecha de ingreso • Hora de ingreso • Número Inmueble
2	Listado de registros	Visualizar en una lista los registros de visitas realizados.	Mostrar una vista de cara al propietario en el que sea fácil visualizar los registros de huéspedes.
3	Eliminar registro	El propietario o arrendatario puede tener acceso a eliminar el registro, en caso de que el visitante ya no pueda asistir.	Incluir un botón de eliminar en el listado de registros.

Tabla 10. HU: Registro de visitante. Fuente: Elaboración propia

 **Appartment** Inicio Salir

Registrar Visitante

Nombre PAULA	Apellido LOPEZ	Tipo Documento CC
Número 1231164345	Apartamento TORRE UNO - A 101	Fecha 15/11/2023 10:30 a. m.
Email paulalopez12@gmail.com	Telefono 3214523445	

[Create](#) [Volver](#)

© 2023 - Appartment.Web - [Privacy](#)

Ilustración 21. Registro de visitante. Fuente: Elaboración Propia



Ilustración 22. Listado de visitantes. Fuente: Elaboración Propia

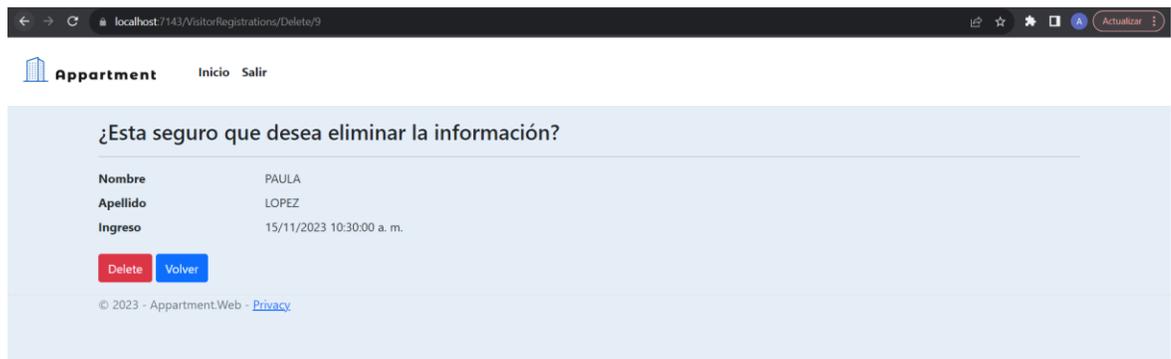


Ilustración 23. Eliminar registro de visitante. Fuente: Elaboración Propia

Historias de Usuario			
No	8	Usuario:	Propietario o arrendatario
Título	Reserva de Zonas Comunes		
Características/Funcionalidad			
Yo como usuario propietario o arrendatario, deseo poder realizar la reserva de zonas comunes de la copropiedad.			
Razón/resultados			
Para hacer un uso adecuado y organizado de las instalaciones compartidas de la copropiedad			
Criterios de Aceptación			
Funcional	X		No Funcional
No	Título	Contexto	Evento
1	Creación de zona común	El administrador debe poder ingresar las zonas comunes disponibles	Incluir una opción de creación de zonas y dar la opción de editar.

2	Visualización de áreas comunes.	Visualizar la ocupación de zonas comunes realizadas por los usuarios	Mostrar un listado con el listado de fechas reservadas
3	Confirmación	Validar si la reserva se registró exitosamente	Se debe mostrar un mensaje de confirmación de la reserva y aclarar que los cargos asociados al pago de la reserva serán cobrados en la próxima cuota de administración.
4	Eliminar registro	Los usuarios pueden tener acceso a eliminar la reserva, en caso de no usar el espacio	Incluir un botón de eliminar en el listado de registros.

Tabla 11. HU: Reserva de Zonas Comunes. Fuente: Elaboración propia

The screenshot shows a web browser window at localhost:7143/Zones/Create. The page title is 'Appartment' with navigation links for 'Inicio' and 'Salir'. The main heading is 'Crear Zonas'. Below it is a form with a text input field labeled 'Zona', a checkbox labeled 'BRecordStatus', and two buttons: 'Crear' and 'Volver'. At the bottom, there is a footer with '© 2023 - Appartment.Web - Privacy'.

Ilustración 24. Crear zona común. Fuente: Elaboración Propia

The screenshot shows a web browser window at localhost:7143/Zones. The page title is 'Appartment' with navigation links for 'Inicio' and 'Salir'. The main heading is 'Zonas' with a 'Crear' button. Below is a table listing zones:

Zona	Fecha Creación	Fecha Modif	Activo	
SALON COMUNAL UNO	24/09/2023 9:03:23 p. m.	1/11/2023 11:09:55 p. m.	<input checked="" type="checkbox"/>	Editar
SALON COMUNAL DOS	21/10/2023 10:56:50 p. m.	21/10/2023 10:56:50 p. m.	<input checked="" type="checkbox"/>	Editar
GIMNASIO	30/10/2023 8:32:57 p. m.	30/10/2023 8:32:57 p. m.	<input checked="" type="checkbox"/>	Editar

At the bottom, there is a footer with '© 2023 - Appartment.Web - Privacy'.

Ilustración 25. Listado de zonas. Fuente: Elaboración Propia



Ilustración 26. Crear reserva de zona común. Fuente: Elaboración Propia

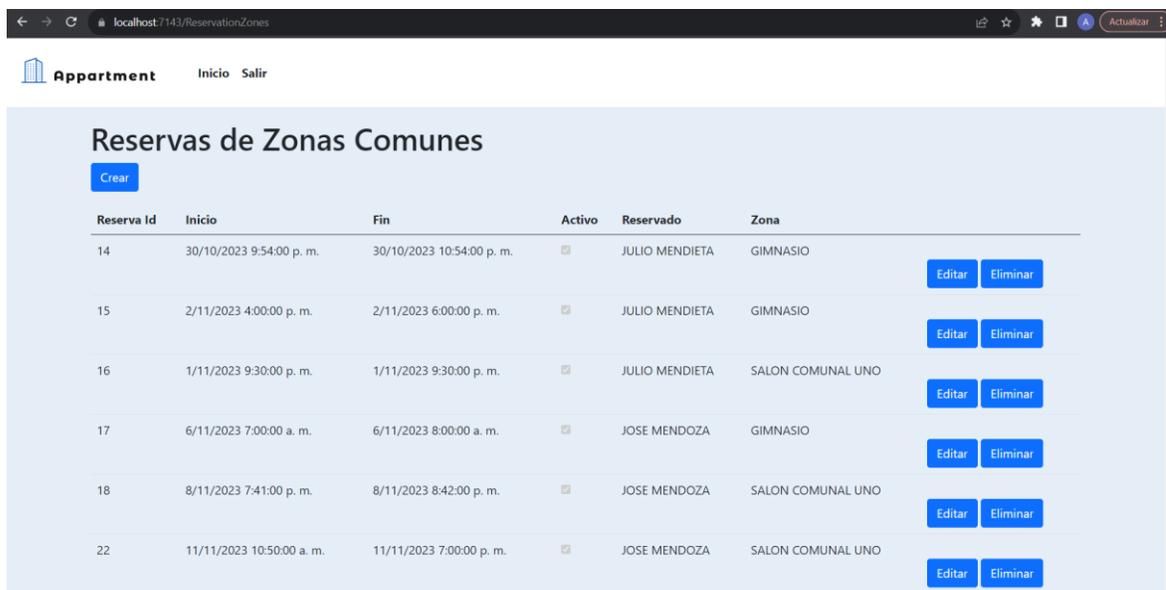


Ilustración 27. Registro de zonas comunes. Fuente: Elaboración Propia

4.3. Requerimientos No Funcionales

A continuación, se procede a hacer el listado de los requerimientos no funcionales a tener en cuenta sobre el proyecto.

- La aplicación web debe ser responsiva.
- La aplicación web debe estar disponible 24/7.
- La interfaz debe ser entendible y fácil de usar para los usuarios.
- La aplicación debe mostrar mensajes de error en los casos en los que se diligencien mal los datos.
- La aceptación de términos y condiciones será un proceso externo a la aplicación el cual debe ser solicitado por la administración.

4.4. Diseño y arquitectura

De acuerdo con los requerimientos establecidos y las necesidades de los usuarios, se optó por realizar una arquitectura MVC (Modelo-Vista-Controlador), con el fin de segmentar e identificar de una manera más fácil los datos e interfaz que se requieren para atender este tipo de gestiones.

Así mismo, se tiene en cuenta el posible crecimiento que puede tener la aplicación en cuanto a usuarios y funcionalidades, para que de esta manera sea más fácil aplicar opciones de microservicios al sistema.

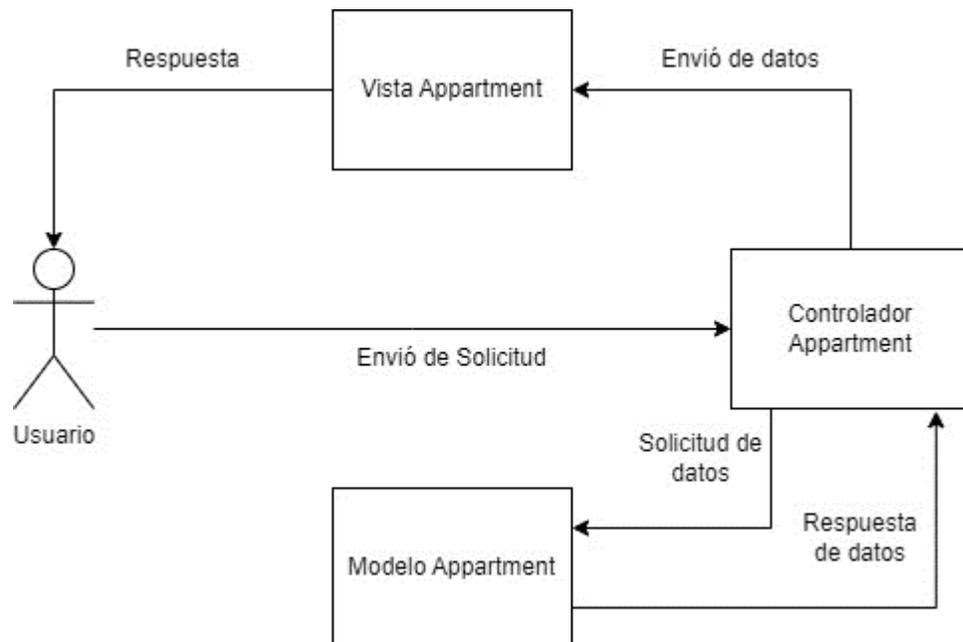


Ilustración 28. Diseño y arquitectura. Fuente: Elaboración Propia

En la anterior imagen se logra evidencia como el usuario inicia enviando las solicitudes requeridas al controlador, para que posteriormente el sistema realice

las validaciones y lógica de las solicitudes que corresponda y posteriormente remitir los datos al modelo y esperar la respuesta a la solicitud realizada; retornando al controlador y redireccionando a la vista para finalmente ser presentado el resultado al usuario.

4.5. Diagrama de despliegue

El diagrama que se expone a continuación muestra como los diferentes componentes instalados en el sistema de información se comunican entre sí en el momento de ejecución, con el fin de proporcionar una experiencia sencilla y coherente al momento de desarrollar.

La aplicación principalmente se va a desarrollar en el framework de .Net Core versión 7, por medio de Visual Studio, cuya base de datos estará montada en un servidor de SQL server 2019 la cual almacenará toda la información de registros a realizar por parte de los usuarios de la copropiedad.

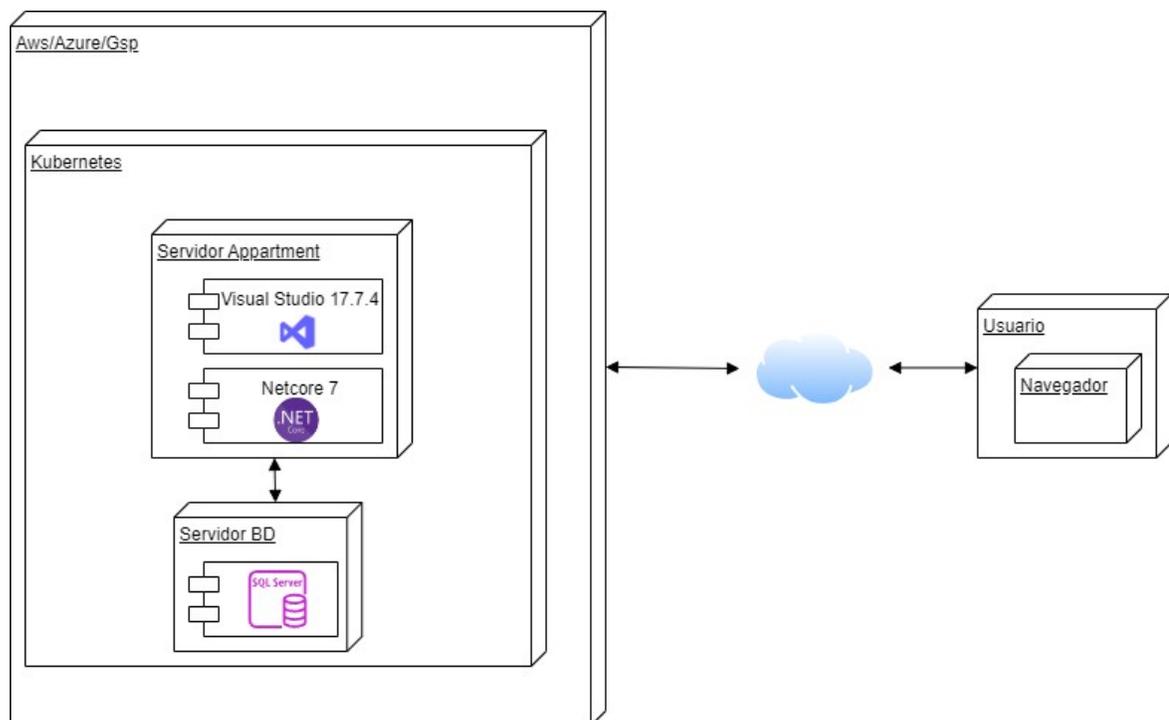


Ilustración 29. Diagrama de despliegue. Fuente: Elaboración Propia

4.6. Diagrama de secuencia

A continuación, se presentan dos diagramas de secuencia enfocados en mostrar cómo se realizarían las actividades principales realizadas en la aplicación, las cuales se basan en realizar el registro de visitas y zonas comunes por parte de los diferentes usuarios de la copropiedad.

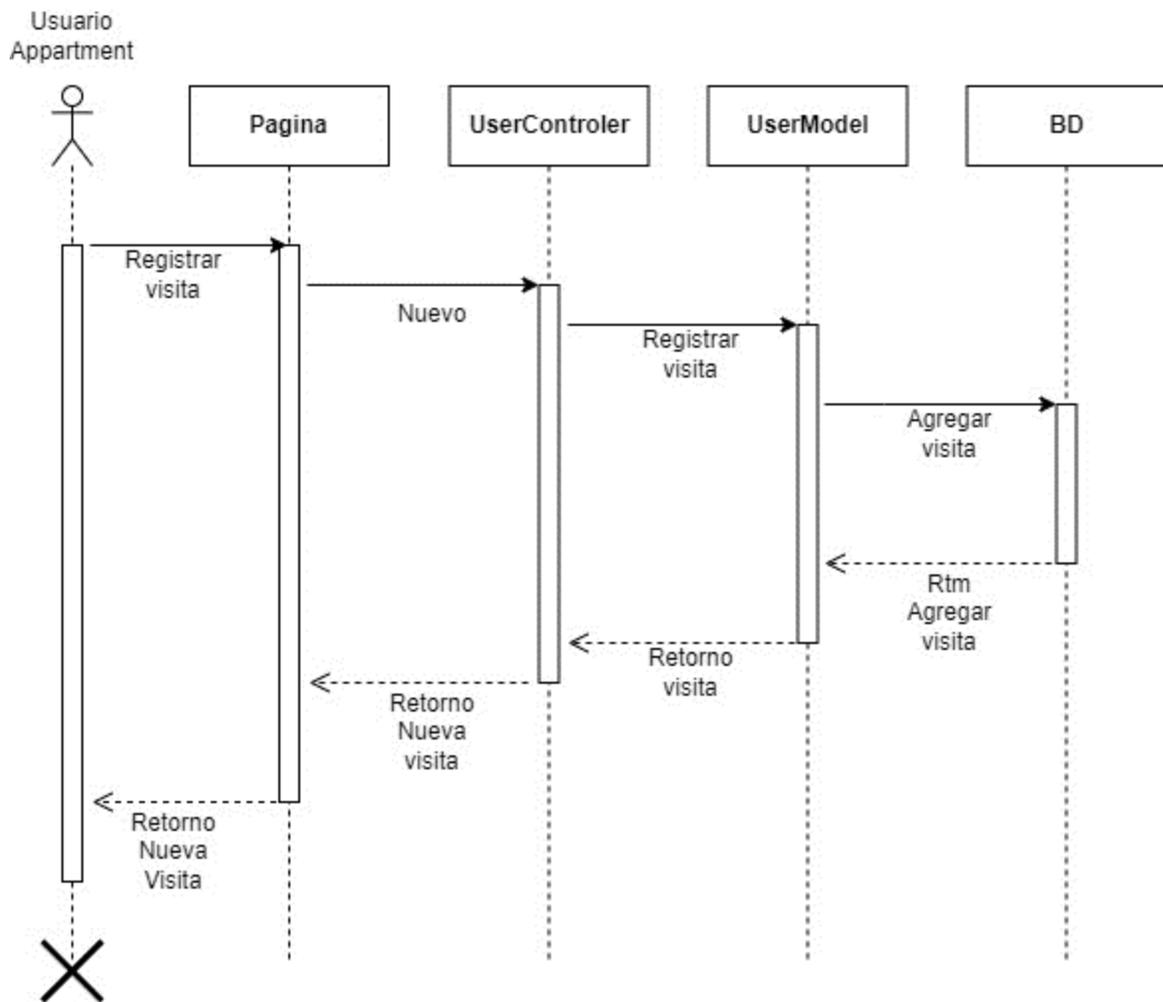


Ilustración 30. Diagrama de secuencia 1. Fuente: Elaboración Propia

La secuencia principal se enfoca en llevar la solicitud de registros realizada por parte de usuario al controlador, modelo y posteriormente a la base de datos, con el fin de retornar el nuevo registro al sistema.

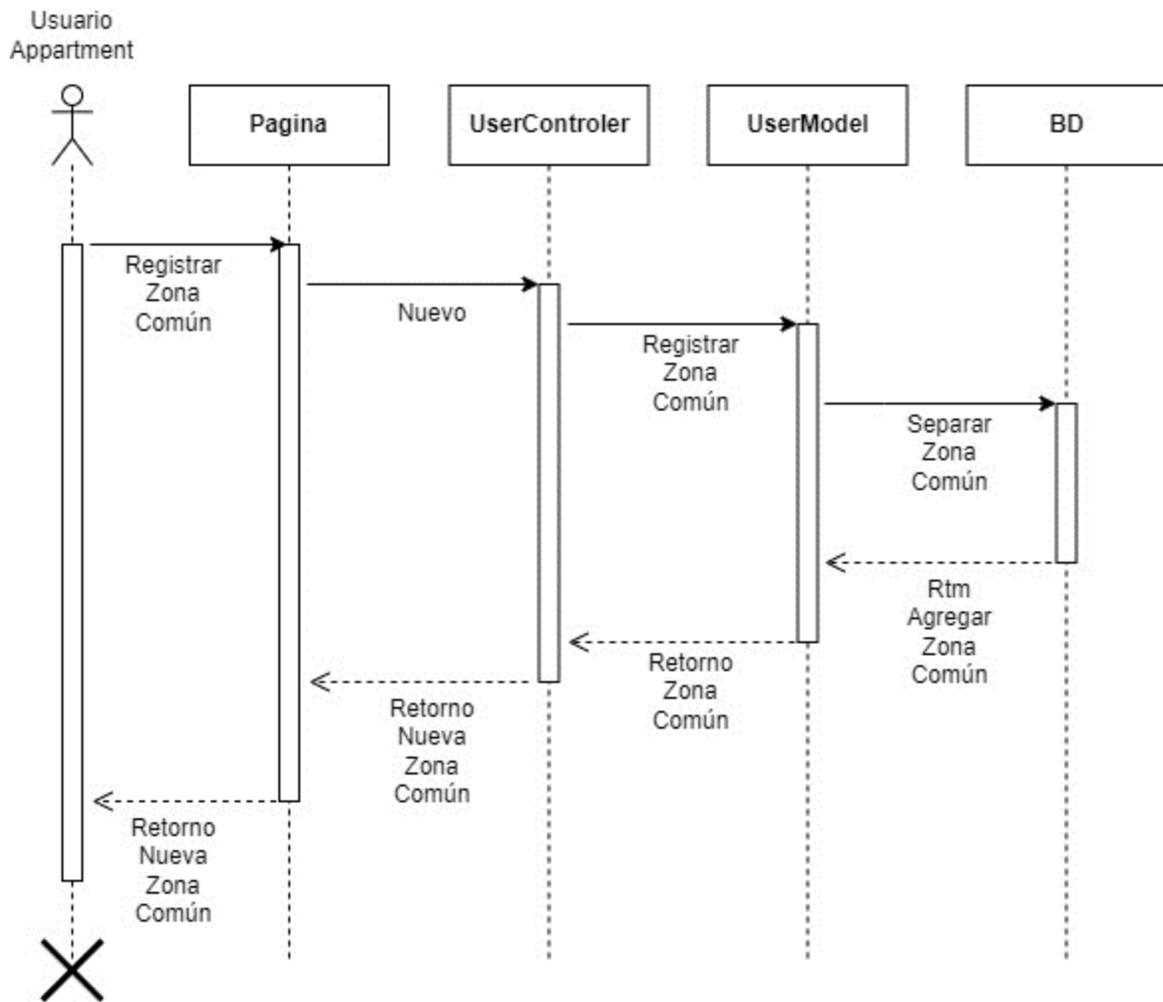


Ilustración 31. Diagrama de secuencia 2. Fuente: Elaboración Propia

4.7. Diagrama de clases

De acuerdo con los lineamientos definidos en el diseño de datos y dando prioridad a la relación que se presenta en el modelo relacional. A continuación, se presenta un diagrama de clases en el que se muestra de una manera más fácil la forma en la que los datos serán almacenados y cómo se han decidido organizar para poder acceder a ellos de una manera más efectiva.

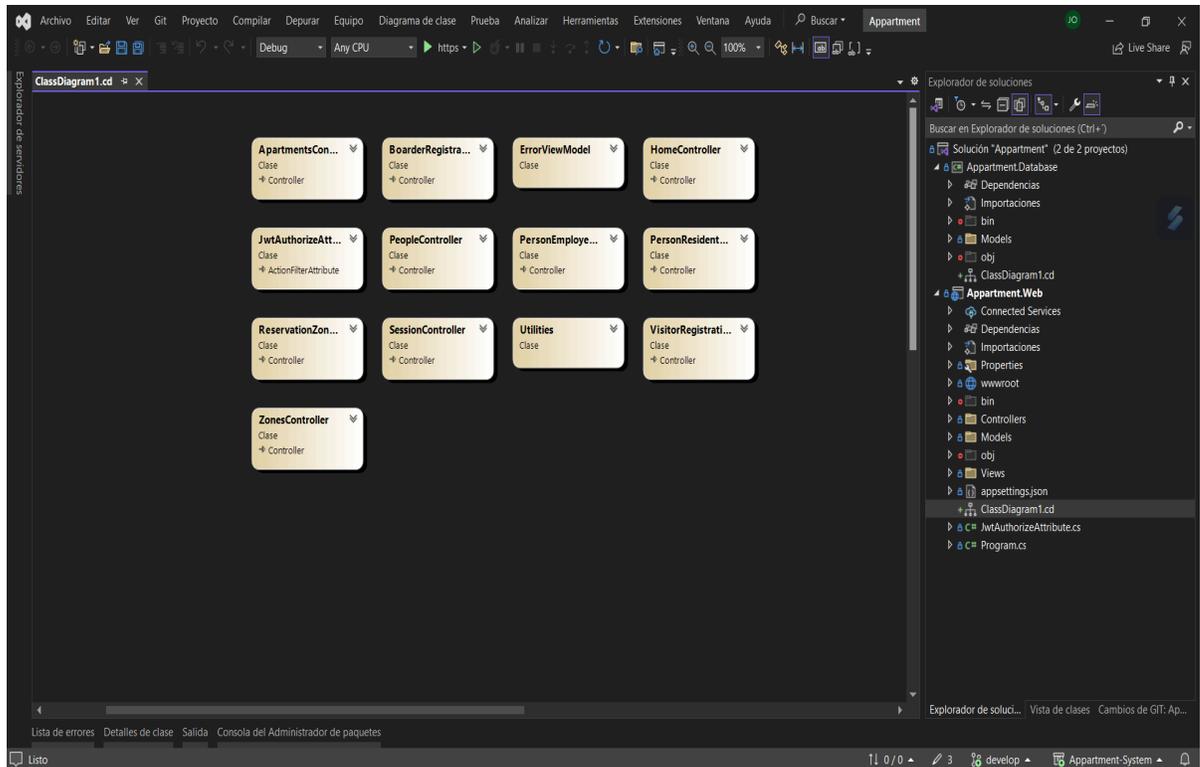


Ilustración 32. Diagrama de clases 1. Fuente: Elaboración Propia

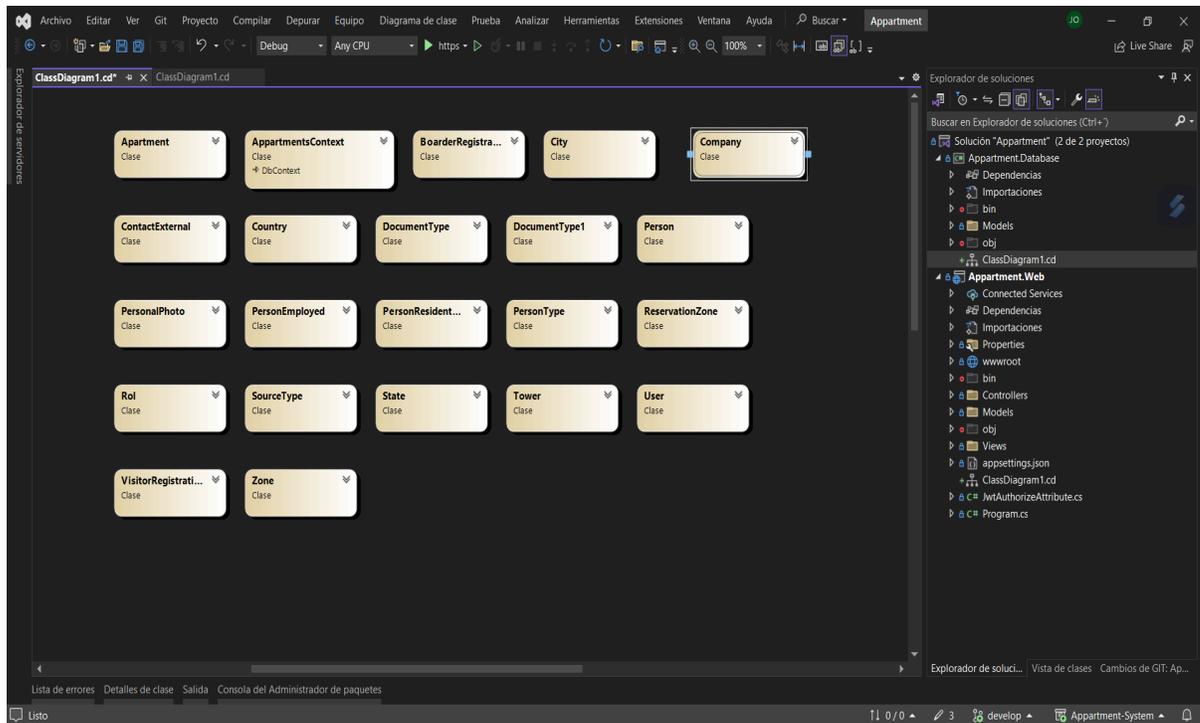


Ilustración 33. Diagrama de clases 2. Fuente: Elaboración Propia

4.8. Arquitectura de alto nivel

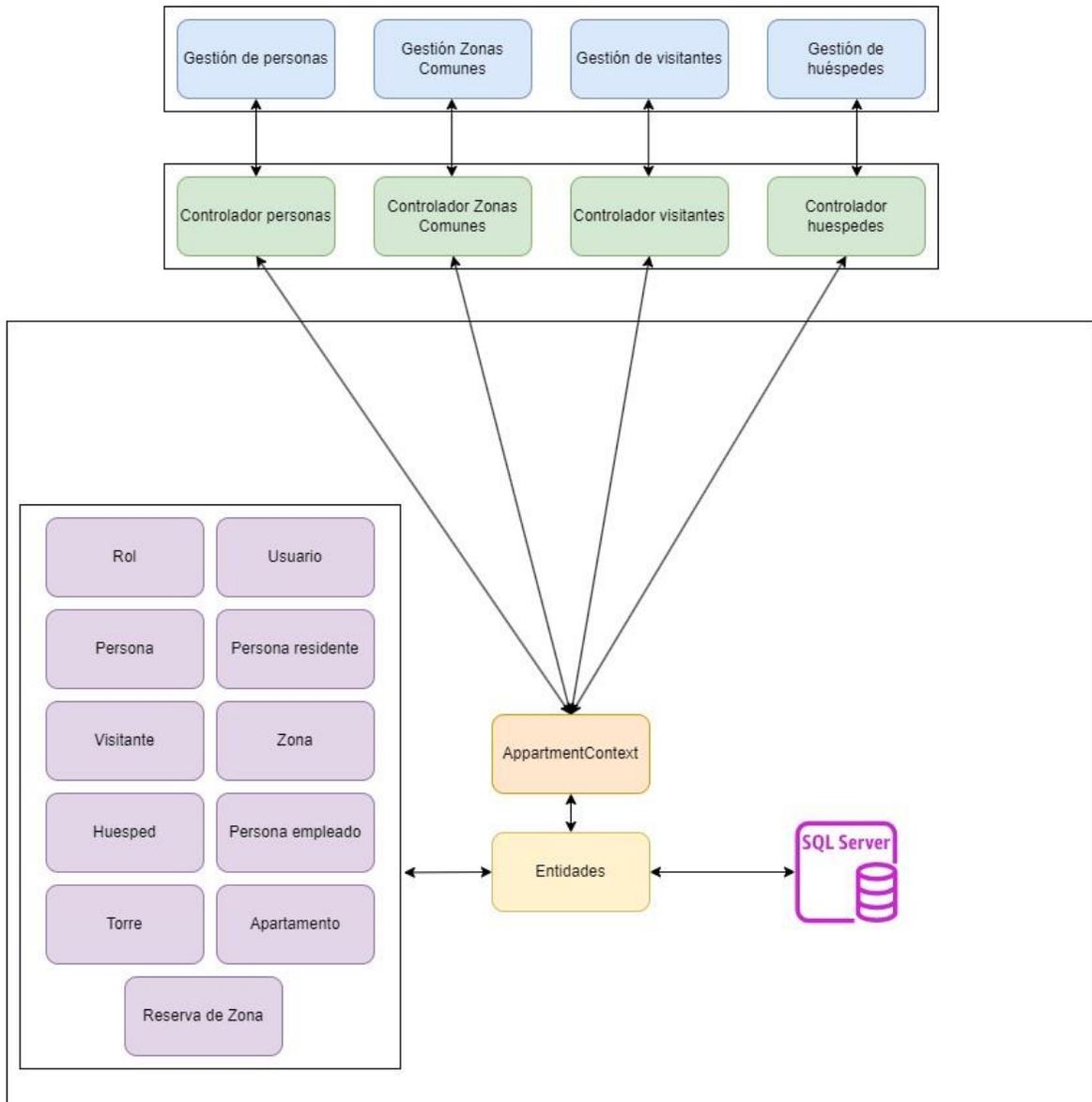


Ilustración 34. Arquitectura de alto nivel. Fuente: Elaboración Propia

5. Construcción

Para la construcción de la aplicación web desarrollada se decidió optar por brindar a los administradores y usuarios de copropiedad una plataforma digital que facilite la gestión de procesos y mejore la interacción con la propiedad. Para lograr dicho objetivo, se procedió a realizar una arquitectura sólida y segura que incluye el backend creado en .NET 7 con Visual Studio y una base de datos alojada en SQL Server 2019. De igual manera el frontend se ha diseñado de manera responsive para garantizar una experiencia de usuario que sea optima en diferentes dispositivos.

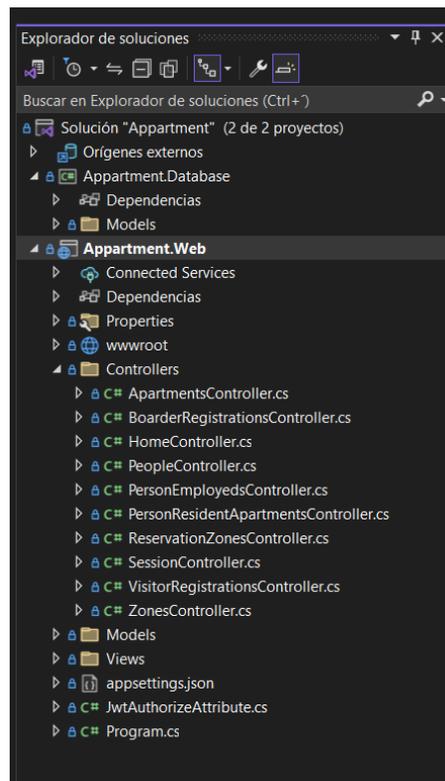


Ilustración 35. Capas implementadas a nivel de código. Fuente: Elaboración Propia

El backend de la aplicación alberga la lógica principal de los servicios que ofrece Appartment e incluye funciones importantes para el control de visitantes y la reserva de

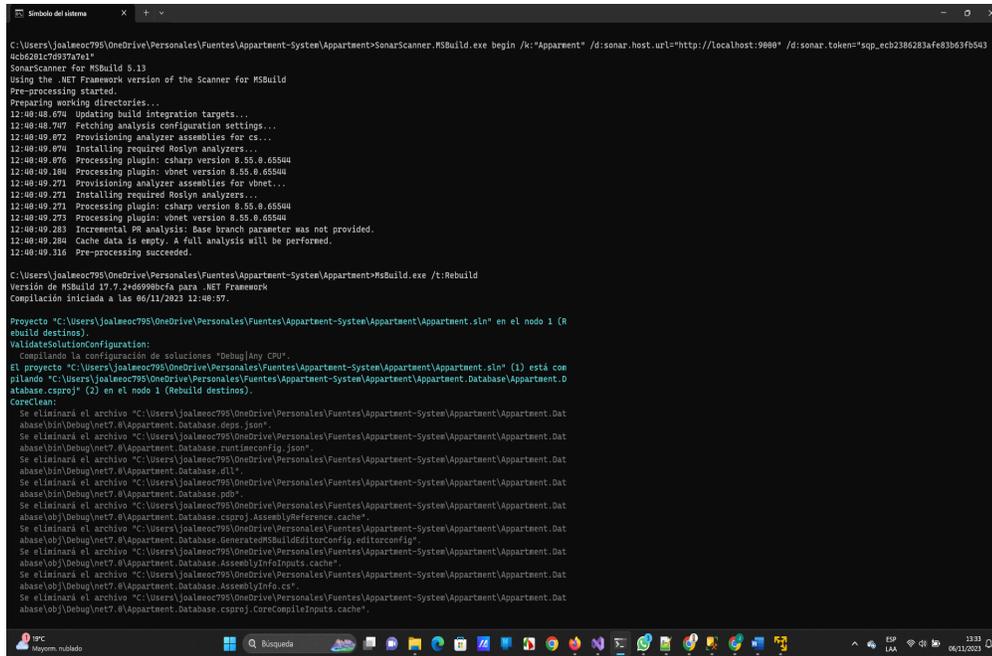
zonas comunes. Para permitir un almacenamiento seguro y eficiente de datos, la base de datos de SQL Server 2019 se ha integrado estrechamente con el backend. Esta seguridad es de suma importancia y se implementaron procesos de autenticación y autorización para garantizar la privacidad de los datos y la protección de la aplicación contra accesos no autorizados.

Para el frontend de la aplicación crea una interfaz de usuario fácil de usar y responsive utilizando tecnologías web convencionales como HTML y CSS. Esto permite a los usuarios interactuar con la aplicación de manera fácil y eficiente, independientemente del dispositivo.

6. Pruebas

6.1. Pruebas de proceso

Se procede a lanzar el escaneo por medio de SonarQ Local a fin de poder obtener los resultados o hallazgos resultantes del análisis de código estático.



```
C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System>SonarScanner.MSBuild.exe begin /k:"Appartment" /d:sonar.host.url="http://localhost:9000" /d:sonar.token="sup_ech2386283afe8363f05434cb2621c79397a7e1"
SonarScanner for MSBuild 5.13
Using the .NET Framework version of the Scanner for MSBuild
Pre-processing started.
Preparing working directories...
12:48:48.674 Updating build integration targets...
12:48:48.747 Fetching analysis configuration settings...
12:48:48.872 Provisioning analyzer assemblies for cs...
12:48:48.874 Installing required Roslyn analyzers...
12:48:48.876 Processing plugin: csharp version 8.55.0.65544
12:48:48.184 Processing plugin: vbnet version 8.55.0.65544
12:48:48.271 Provisioning analyzer assemblies for vbnet...
12:48:48.271 Installing required Roslyn analyzers...
12:48:48.271 Processing plugin: csharp version 8.55.0.65544
12:48:48.271 Processing plugin: vbnet version 8.55.0.65544
12:48:48.283 Incremental PR analysis: Base branch parameter was not provided.
12:48:48.284 Cache data is empty. A full analysis will be performed.
12:48:48.316 Pre-processing succeeded.

C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System>Appartment>MSBuild.exe /t:Rebuild
Versión de MSBuild 17.7.24069906cfa para .NET Framework
Compilación iniciada a las 06/11/2023 12:48:37.

Proyecto "C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment\Appartment.sln" en el nodo 1 (Rebuild destinos).
ValidatesSolutionConfiguration:
  Compilando la configuración de soluciones "Debug|Any CPU".
  El proyecto "C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment\Appartment.sln" (1) está con pilando "C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment\Appartment.Database\Appartment.Database.csproj" (2) en el nodo 1 (Rebuild destinos).
CoreClean:
  Se eliminará el archivo "C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment\Appartment.Database\bin\Debug\net7.0\Appartment.Database.deps.json".
  Se eliminará el archivo "C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment\Appartment.Database\bin\Debug\net7.0\Appartment.Database.runtimeconfig.json".
  Se eliminará el archivo "C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment\Appartment.Database\bin\Debug\net7.0\Appartment.Database.dll".
  Se eliminará el archivo "C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment\Appartment.Database\bin\Debug\net7.0\Appartment.Database.pdb".
  Se eliminará el archivo "C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment\Appartment.Database\obj\Debug\net7.0\Appartment.Database.csproj.AssemblyReference.cache".
  Se eliminará el archivo "C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment\Appartment.Database\obj\Debug\net7.0\Appartment.Database.generator.build.targets.editorconfig".
  Se eliminará el archivo "C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment\Appartment.Database\obj\Debug\net7.0\Appartment.Database.generator.build.targets.cache".
  Se eliminará el archivo "C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment\Appartment.Database\obj\Debug\net7.0\Appartment.Database.generator.build.targets.cache".
  Se eliminará el archivo "C:\Users\joaleoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment\Appartment.Database\obj\Debug\net7.0\Appartment.Database.generator.build.targets.cache".
```

Ilustración 36. Pruebas SonarQube 1. Fuente: Elaboración Propia

```
Símbolo del sistema x + -
INFO: Sensor C# [csharp]
INFO: Importing results from 6 proto files in 'C:\Users\joalmeoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment.sonarqube\out\1\output-cs'
INFO: Importing 2 Maven reports
INFO: Found 2 MSBuild CI projects: 2 MAIN projects.
INFO: Sensor C# [csharp] (done) | time=457ms
INFO: Sensor Analysis Warnings report [csharp]
INFO: Sensor Analysis Warnings report [csharp] (done) | time=9ms
INFO: Sensor C# File Caching Sensor [csharp]
INFO: Sensor C# File Caching Sensor [csharp] (done) | time=9ms
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=38ms
INFO: SCM Publisher SCM provider for this project is: git
INFO: SCM Publisher 119 source files to be analyzed
INFO: SCM Publisher 119/119 source files have been analyzed (done) | time=484ms
INFO: CPD Executor 8 files had no CPD blocks
INFO: CPD Executor Calculating CPD for 88 files
INFO: CPD Executor CPD calculation finished (done) | time=128ms
INFO: Analysis report generated in 242ms, dir size=957.9 kb
INFO: Analysis report compressed in 362ms, zip size=361.5 kb
INFO: Analysis report uploaded in 96ms
INFO: ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=Appartment
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=Avulukv7F0pg1Anuh6b
INFO: Analysis total time: 58.912 s
INFO:
INFO: EXECUTION SUCCESS
-----
INFO: Total time: 55.799s
INFO: Final Memory: 159/60M
INFO:
The SonarScanner CLI has finished
12:42:22.988 Post-processing succeeded.
C:\Users\joalmeoc795\OneDrive\Personales\Fuentes\Appartment-System\Appartment
```

Ilustración 37. Pruebas SonarQube 2. Fuente: Elaboración Propia

Una vez como se muestra en la imagen inmediatamente anterior confirma la finalización de la ejecución podemos visualizar a nivel de la interfaz Web de SonarQube los resultados obtenidos como los reflejan las imágenes seguidamente presentadas.

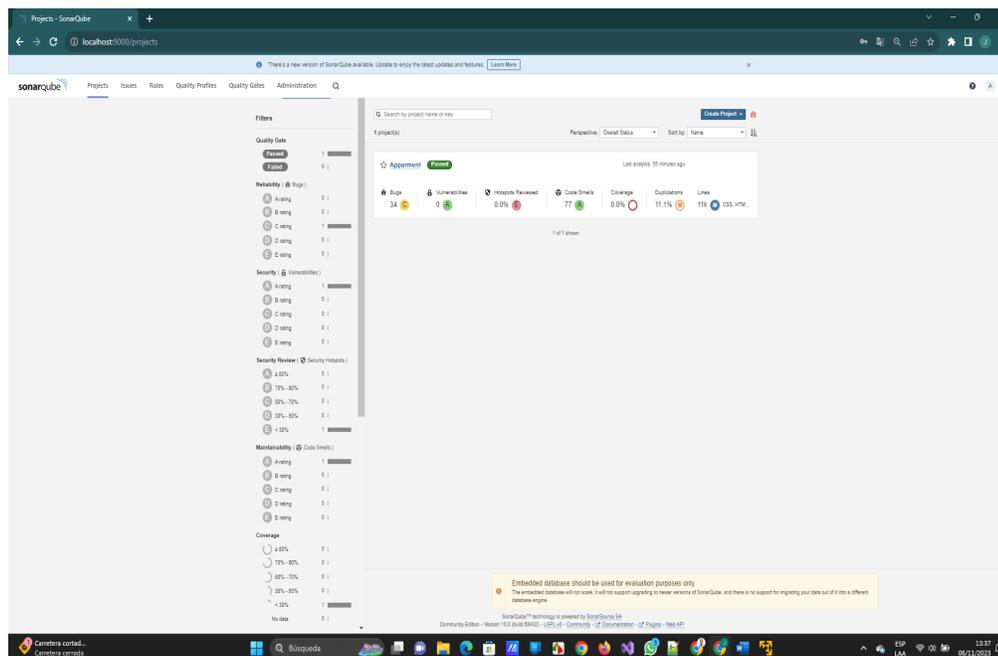


Ilustración 38. Pruebas SonarQube 3. Fuente: Elaboración Propia

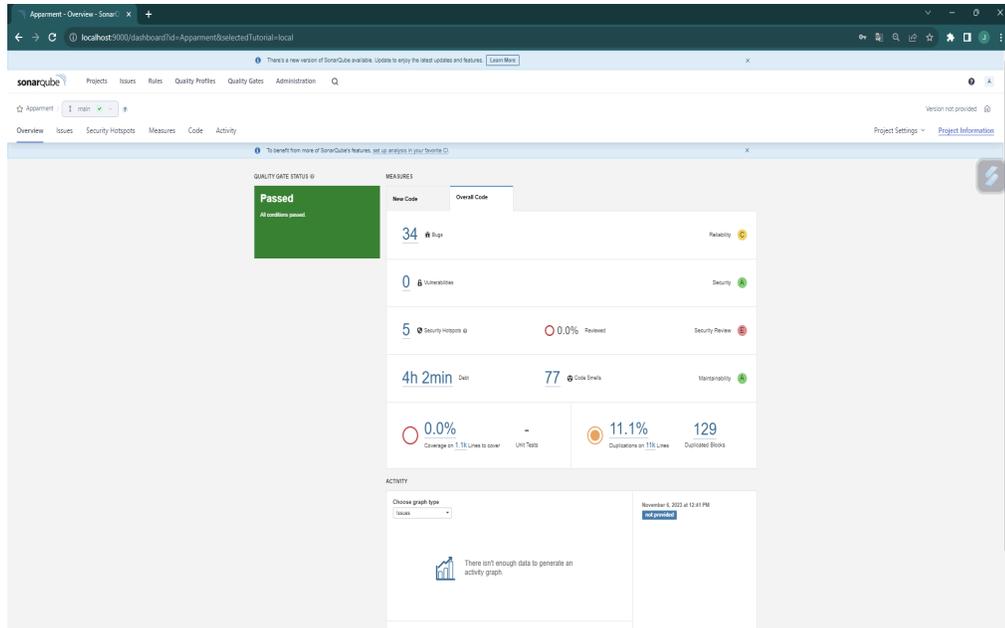


Ilustración 39. Pruebas SonarQube 4. Fuente: Elaboración Propia

Procederemos a revisar y remediar los hallazgos resultantes que representen un riesgo para el buen funcionamiento de nuestro aplicativo.

Posterior a análisis

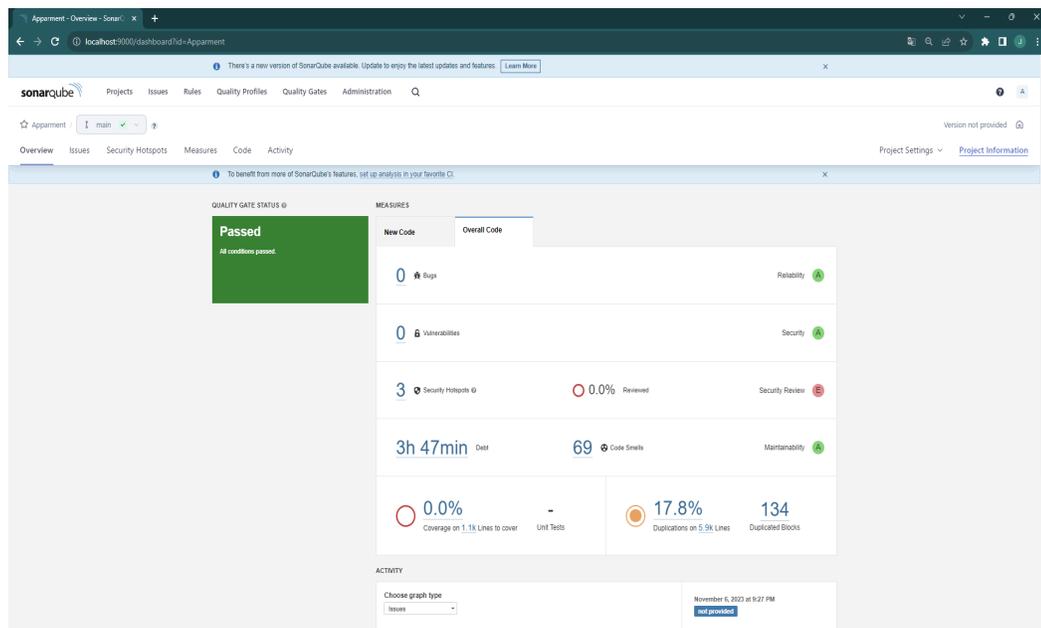


Ilustración 40. Pruebas SonarQube 5. Fuente: Elaboración Propia

Para lograr solucionar los Bugs y Security Hotspots fue necesario hacer unos ajustes a nivel de HTML en las imágenes para adicionar el atributo alt, Quitar una carpeta de estilos CSS que no estaba siendo utilizada por la aplicación y que presentaba inconsistencias en sus hojas de estilos y finalmente validar siempre en ejecución que lo objetos que cargan data antes de usarse se revise que no sean null.

6.1. Pruebas de producto y seguridad

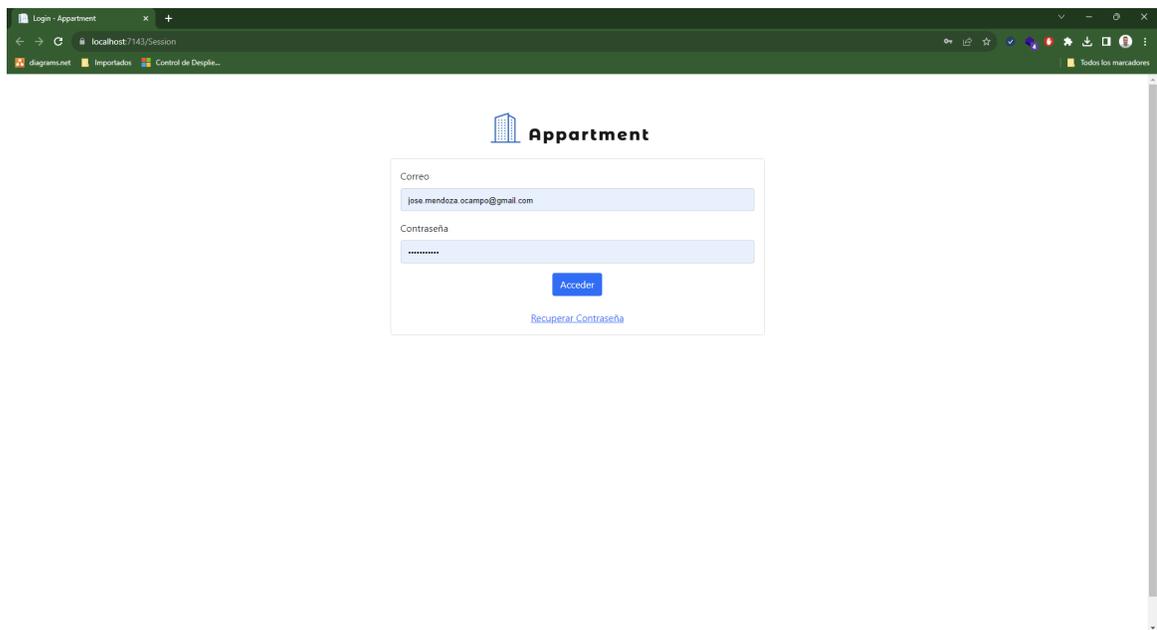


Ilustración 41. Local Host. Fuente: Elaboración Propia

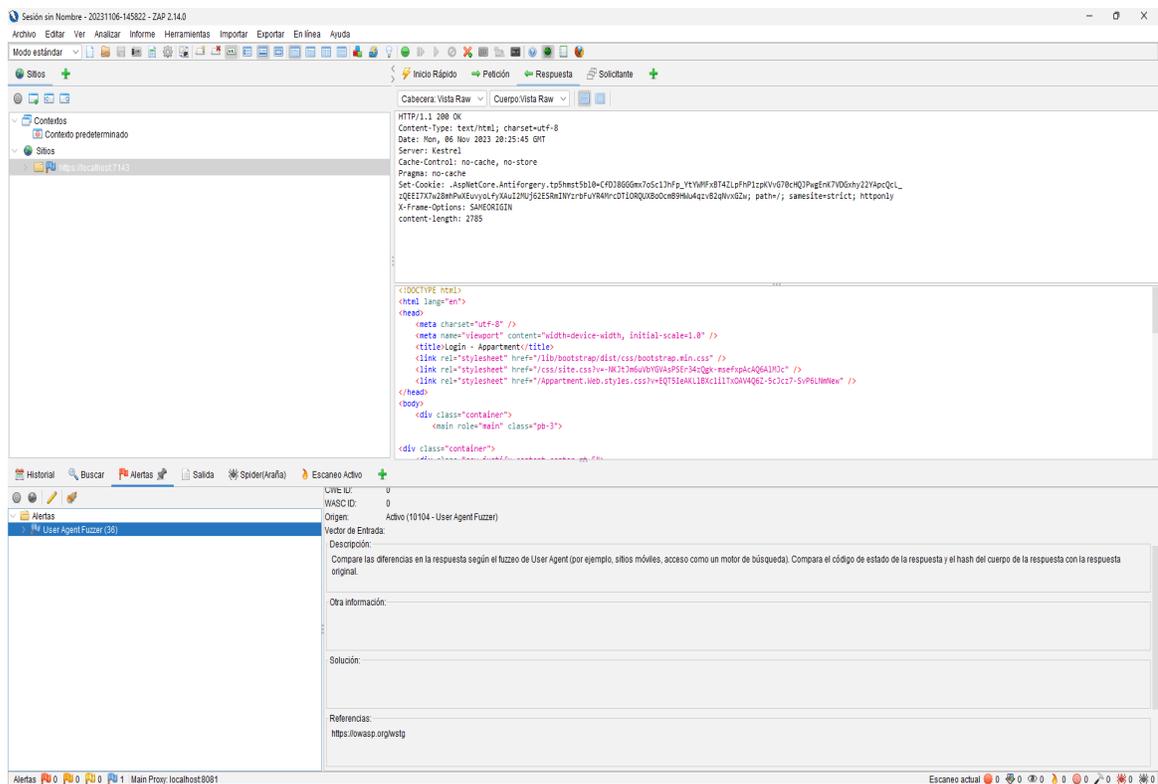


Ilustración 42 Pruebas Owasp Zap. Fuente: Elaboración Propia

La imagen anterior nos permite visualizar el resultado del análisis dinámico efectuado con la aplicación Owasp Zap. Y en donde podemos apreciar que no tenemos hallazgos resultantes del escaneo, gracias a las bondades provistas por el framework de desarrollo que provee los controles para evitar o mitigar posibles vulnerabilidades.

7. Instalación y Configuración

A continuación, se realiza la explicación del paso a paso realizado para poder generar la instalación y contenerización por medio del IDE de Visual Studio .Net y Docker Desktop.

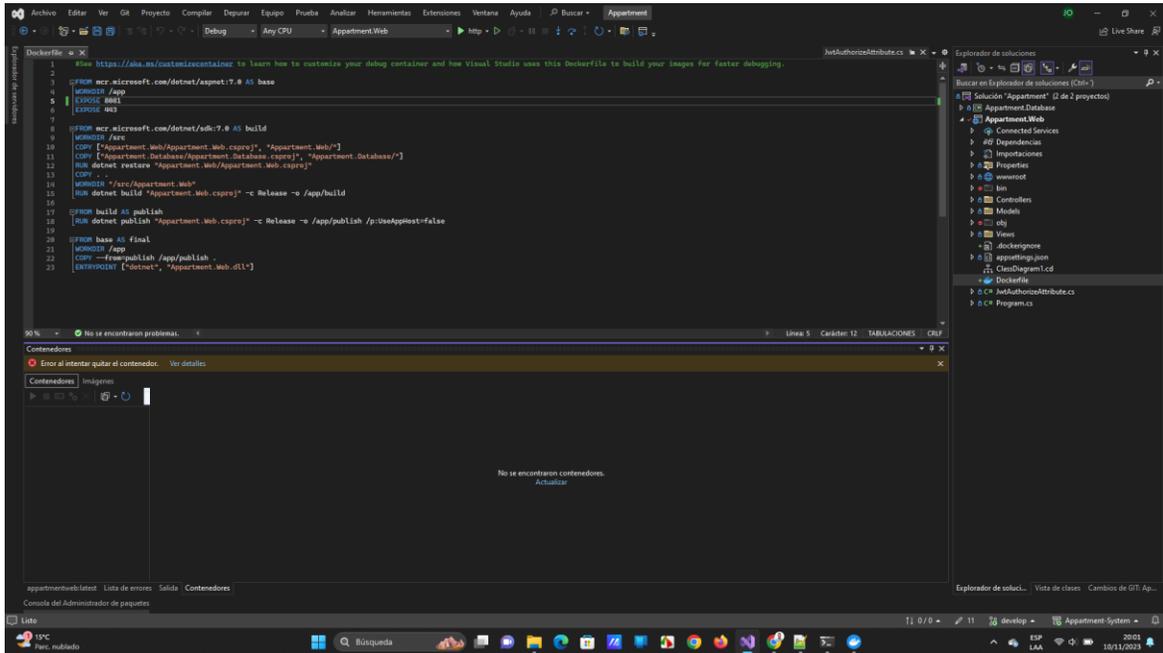


Ilustración 43 Instalación y configuración 1. Fuente: Elaboración Propia

En la anterior imagen, se presenta la ventana de explorador de soluciones en la cual se evidencia el archivo Dockerfile, que contiene la definición para la generación de la imagen que será desplegada a nivel de Docker y los puertos por medio del cual se disponibilizara la aplicación en el contenedor.

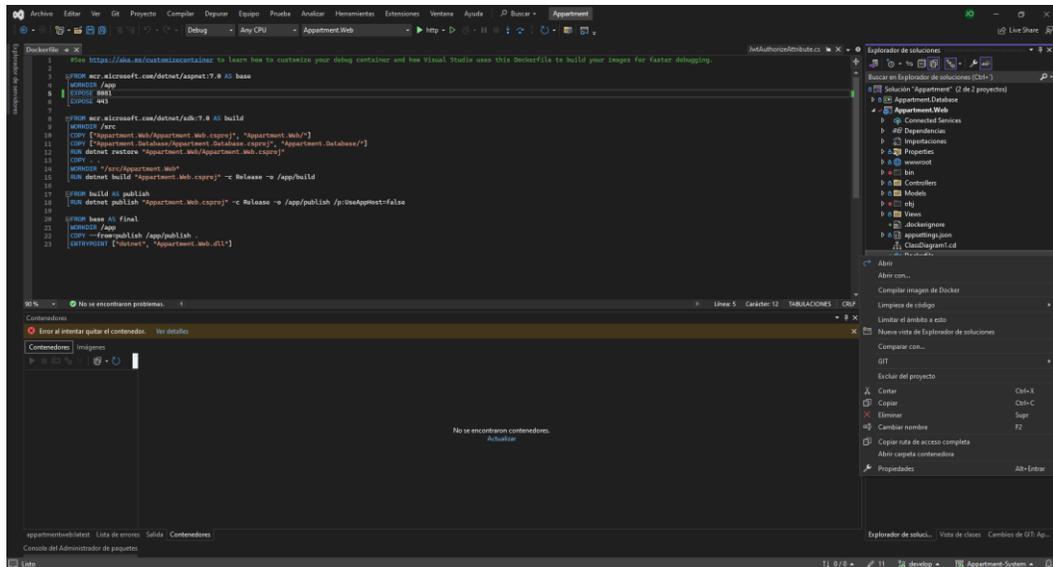


Ilustración 44 Instalación y configuración 2. Fuente: Elaboración Propia

Nos ubicamos sobre el archivo de nombre Dockerfile y presionamos click derecho para que de esta forma se presente el menú contextual.

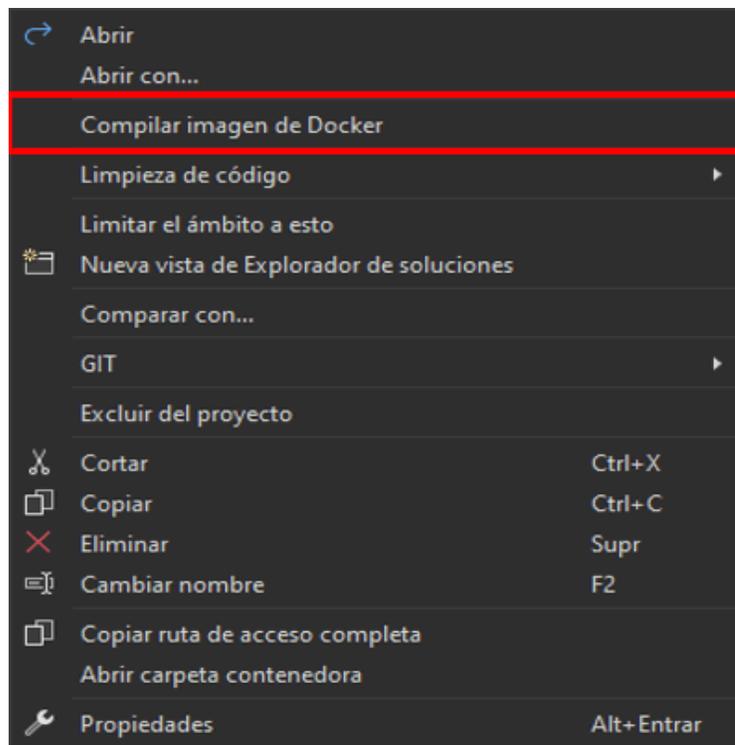


Ilustración 45 Instalación y configuración 3. Fuente: Elaboración Propia

En la imagen anterior, podemos observar las opciones que se presentan en el menú contextual y de las cuales seleccionaremos la de nombre compilar imagen de Docker y de esta forma se procede con la generación de la imagen, obtención de los recurso y compilación de esta.

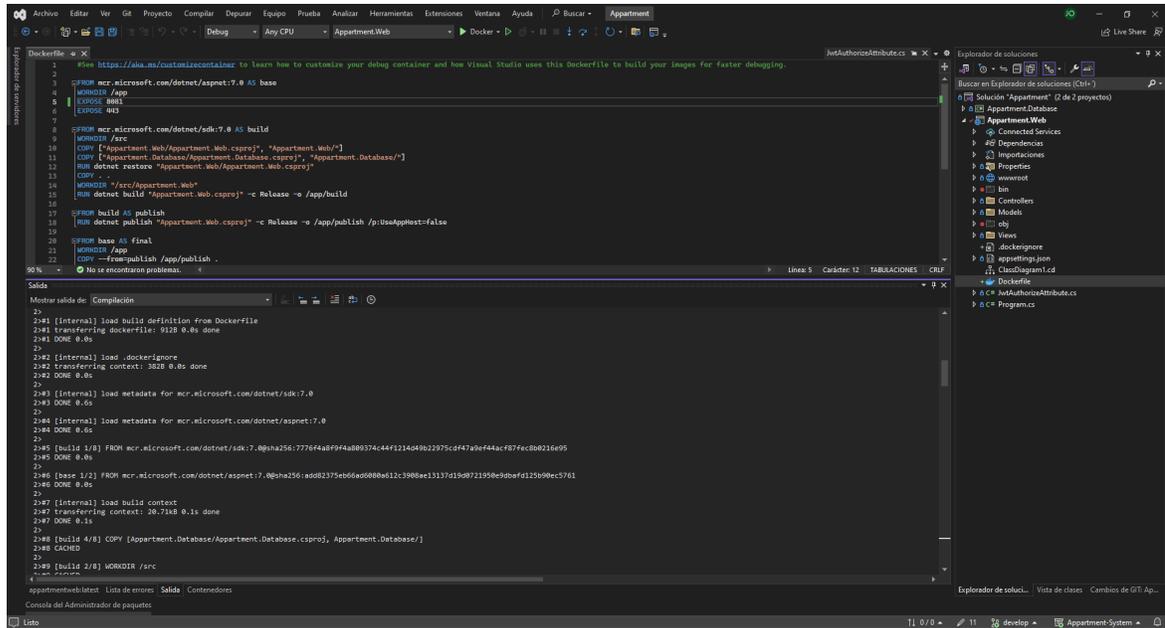


Ilustración 46 Instalación y configuración 4. Fuente: Elaboración Propia

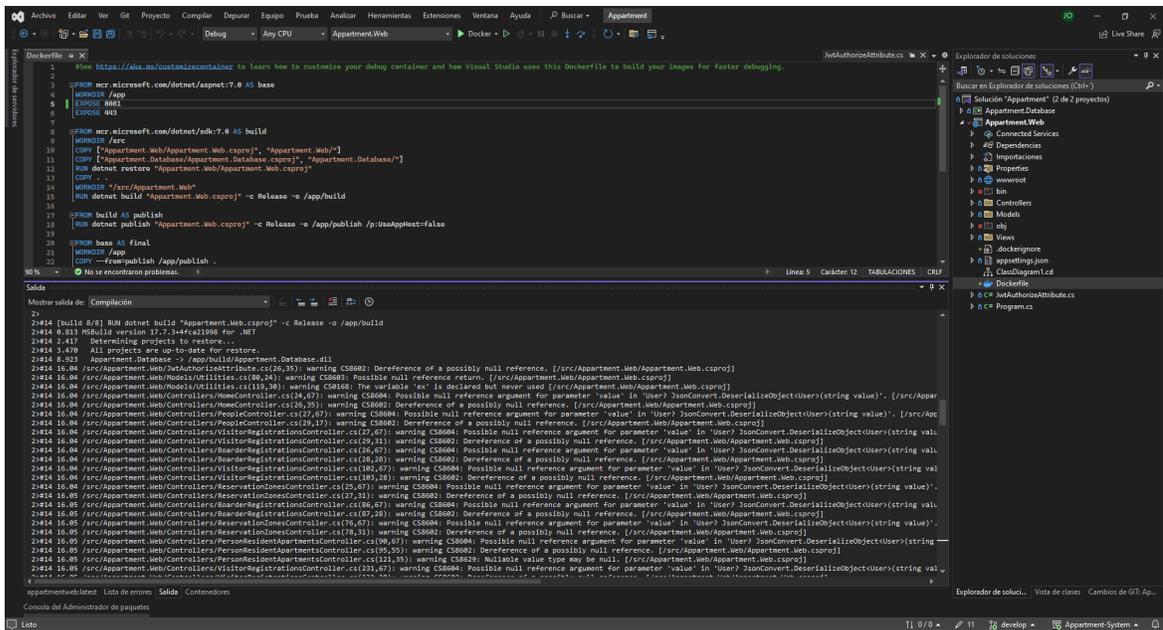


Ilustración 47 Instalación y configuración 5. Fuente: Elaboración Propia

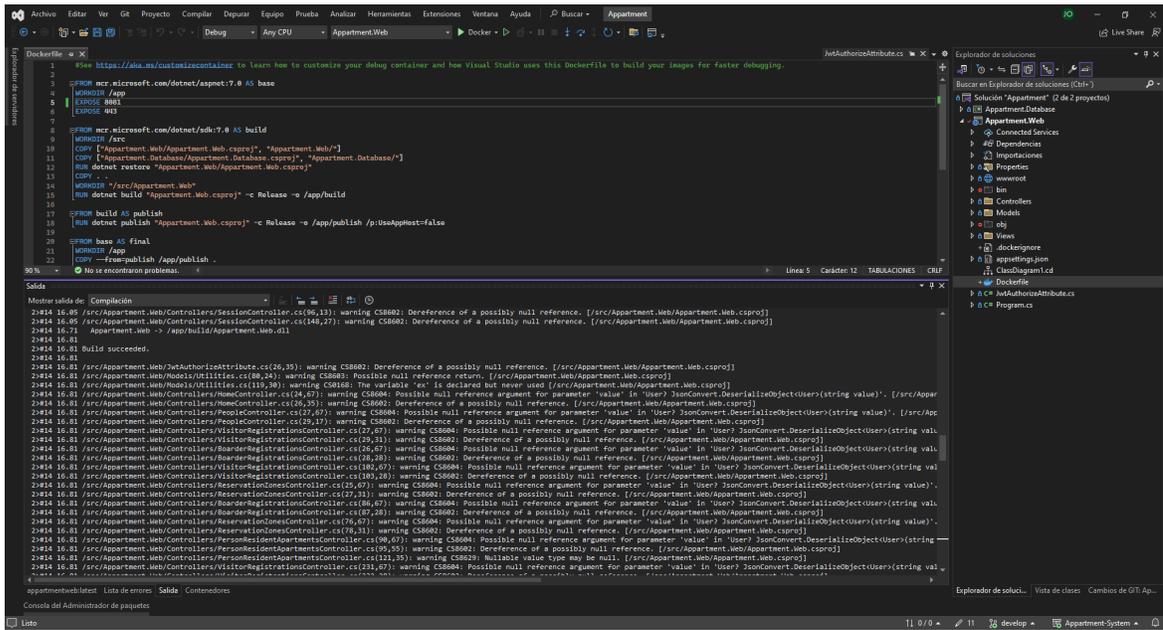


Ilustración 48 Instalación y configuración 6. Fuente: Elaboración Propia

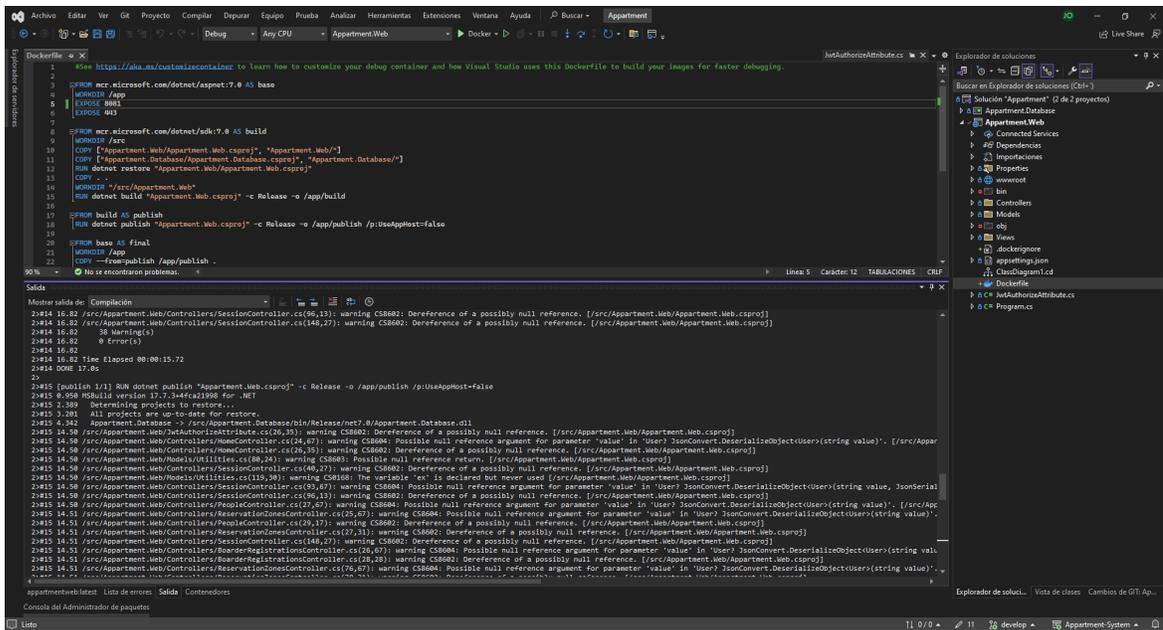


Ilustración 49 Instalación y configuración 7. Fuente: Elaboración Propia

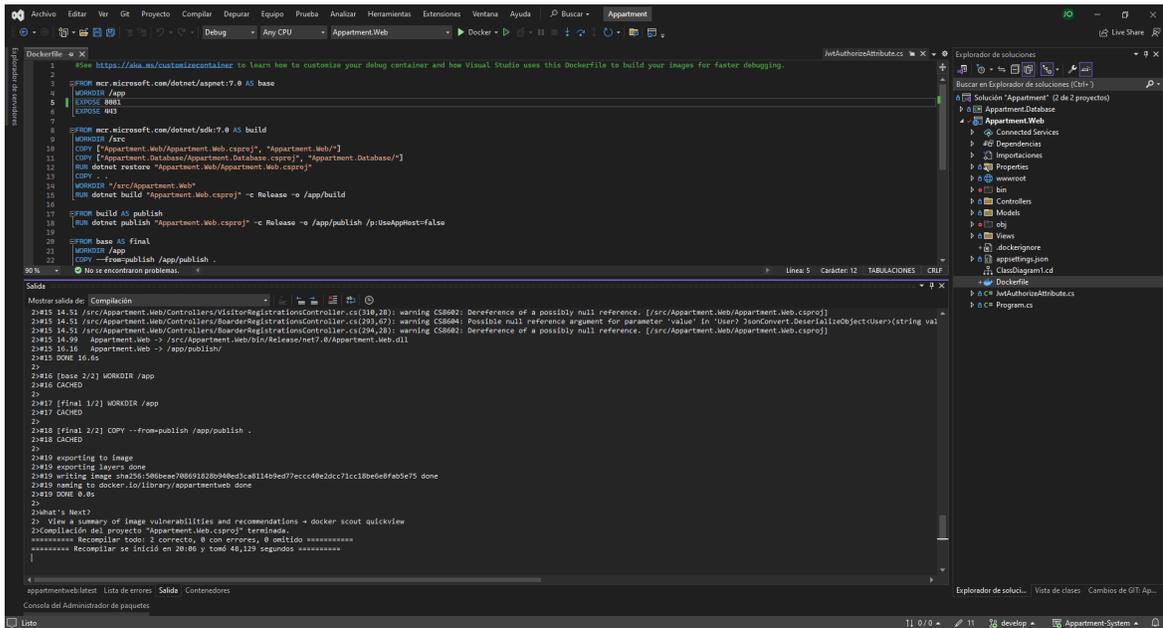


Ilustración 50 Instalación y configuración 8. Fuente: Elaboración Propia

Una vez finalizada la compilación y generación se presenta la confirmación del resultado de la generación al igual que el SHA id correspondiente a la nueva imagen generada y desplegada en el Docker Desktop.

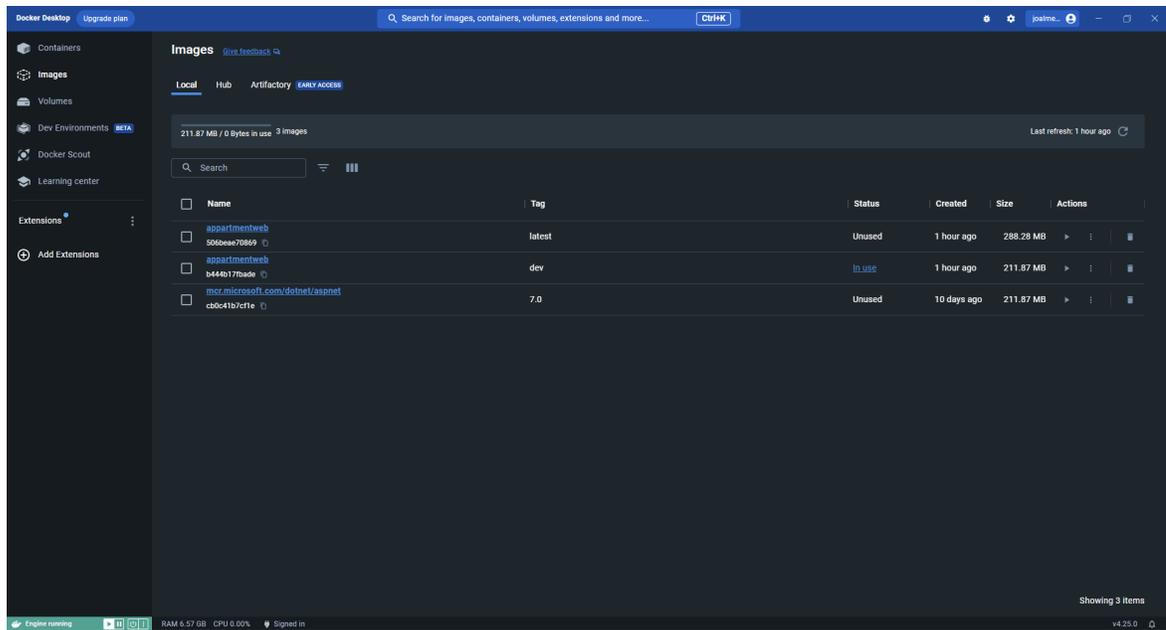


Ilustración 51 Instalación y configuración 9. Fuente: Elaboración Propia

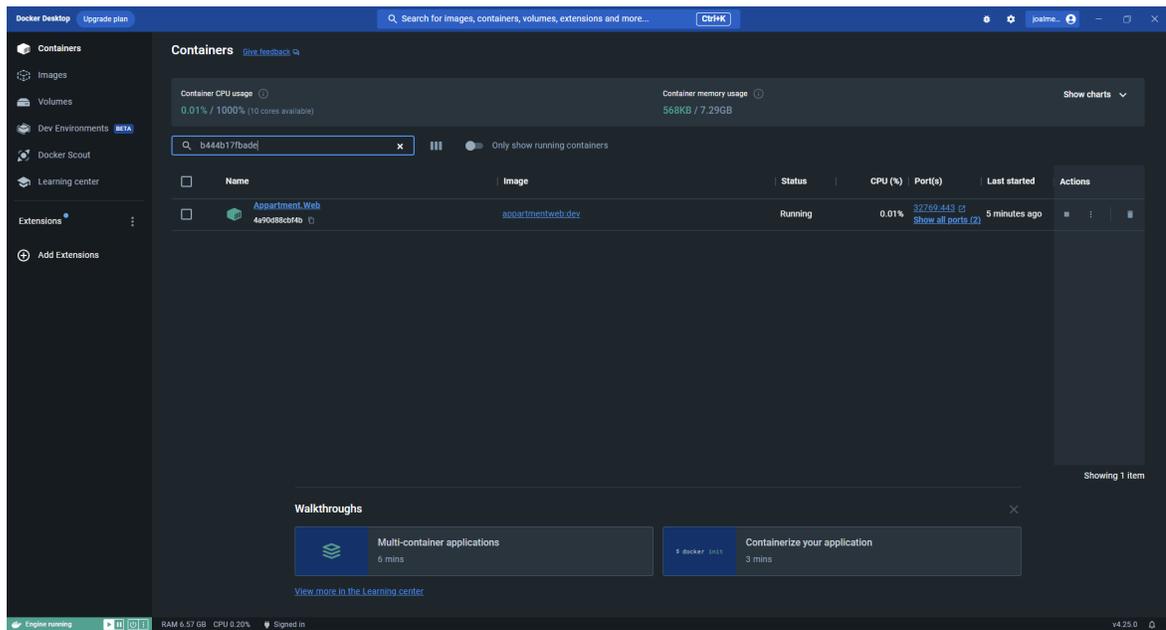


Ilustración 52 Instalación y configuración 10. Fuente: Elaboración Propia

En la imagen anterior podemos observar la imagen dispuesta en Docker Desktop que fue generada y que se encuentra en uso, por medio de la cual ya contamos con el aplicativo publicado para poder hacer uso del mismo desde nuestro navegador.

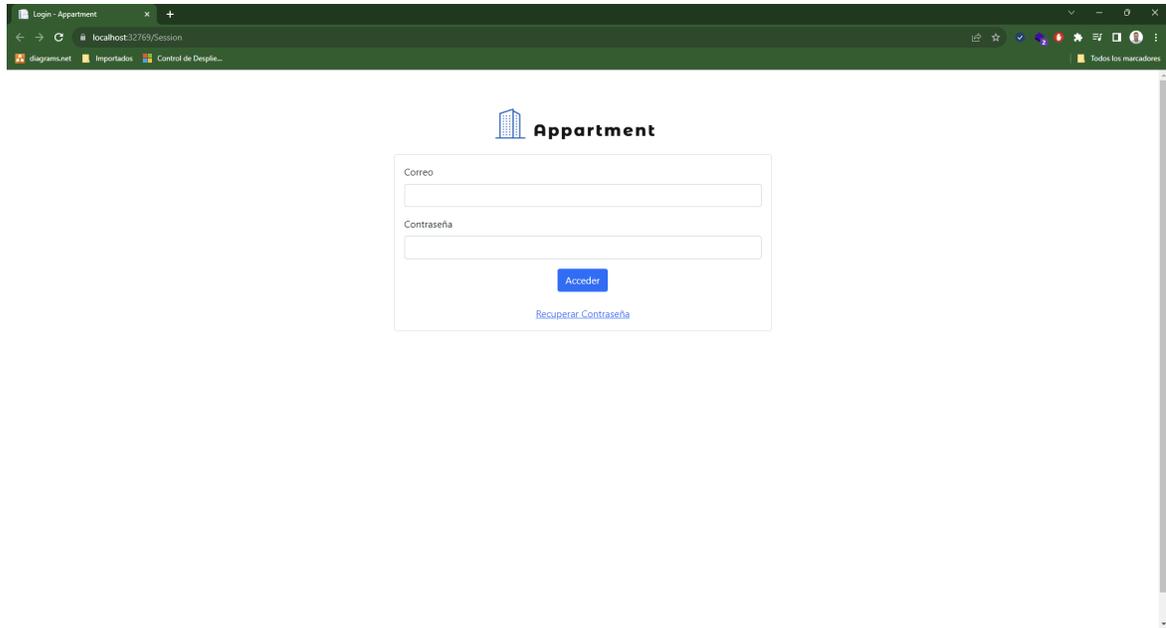


Ilustración 53 Instalación y configuración 11. Fuente: Elaboración Propia

Se procede a abrir la aplicación por medio de un navegador en la que se logra evidenciar que la misma está ejecutándose en el Docker por el puerto 32769.

8. Conclusiones

Se desarrolla y se ejecuta el proyecto cumpliendo con los objetivos generales y específicos planteados en el presente documento, brindando una aplicación Web Responsive que permite a los administradores y usuarios de la copropiedad, optimizar el control de visitantes y reserva de zonas comunes de una manera fácil y segura.

Así mismo, se logró implementar perfiles y usuarios de acceso, con el fin de que la seguridad dentro de las instalaciones sea mucho más robusta y permita a los vigilantes llevar un control más detallado de las personas que ingresan.

Con el desarrollo de la aplicación se logra brindar una solución alternativa a diferentes copropiedades que hoy en día presentan problemas al momento de llevar el control de visitas de personas desconocidas, ayudando a que vigilantes y residentes puedan adaptarse al uso de herramientas tecnológicas de una manera más sencilla evitando procesos que hoy en día se realizan de manera manual.

Durante el proceso se lograron aplicar diferentes conocimientos adquiridos durante el tiempo de estudio de la especialización, dentro de los que se encuentra tácticas para gestión de proyectos, manejo de metodologías ágiles, aplicación de modelos de arquitectura, tácticas para uso de código limpio y seguridad de software.

Ahora bien, pese a que la aplicación actualmente se encuentra enfocada en reserva de zonas comunes y control de visitantes, a futuro cuenta con una gran oportunidad de expansión ya que se pueden implementar nuevos módulos en los que se permita realizar pagos de administración y contar con opciones en las que se puedan atender PQR, haciendo más integral los procesos que se pueden llevar a cabo dentro de las diferentes copropiedades.

Referencias Bibliográficas

Con 209.638 viviendas nuevas vendidas en los últimos 11 meses, el 2021 ya es el mejor año en ventas para Colombia. (s/f). Gov.co. Recuperado el 7 de noviembre de 2023, de <https://minvivienda.gov.co/sala-de-prensa/con-209638-viviendas-nuevas-vendidas-en-los-ultimos-11-meses-el-2021-ya-es-el-mejor-ano-en-ventas-para-colombia>

Apartamentos, Casas y Otros Inmuebles en Venta o Arriendo en Colombia – Metrocuadrado. (s/f). metrocuadrado.com. Recuperado el 7 de noviembre de 2023, de <https://www.metrocuadrado.com/>

Edificios Davivienda. (s/f). Edificiosdavivienda.com. Recuperado el 7 de noviembre de 2023, de <https://www.edificiosdavivienda.com>

Hughes, A., & Stedman, C. (2021, abril 23). *Microsoft SQL Server*. ComputerWeekly.es; TechTarget. <https://www.computerweekly.com/es/definicion/Microsoft-SQL-Server>

Martínez, R. N., Fuentes, J. R. L., Durango, A., & Ramos, D. (2015). *Curso de Ingeniería de Software*. IT Campus Academy.

Pellicer, P. (2021, octubre 21). *¿Qué es el .NET? ¿Para qué sirve?* Blog Emagister; Emagister. <https://www.emagister.com/blog/que-es-el-net-para-que-sirve/>

Propiedata. (s/f). *La plataforma de asambleas exitosas*. Propiedata. Recuperado el 7 de noviembre de 2023, de <https://www.propiedata.com>

¿Qué es SonarQube? (2022, julio 22). *KeepCoding Bootcamps*. <https://keepcoding.io/blog/que-es-sonarqube/>

Sentrio. (2022, agosto 9). *¿Qué es Azure DevOps?* Sentrio. <https://sentrio.io/blog/que-es-azure-devops/>

Software para la Administración de Propiedad Horizontal. (2021, junio 16). Edifito Colombia. <https://edifito.co>

Sommerville, I. (2005). *Ingeniería del software*. Pearson Educación.