UNIVERSIDAD ESTATAL DE BOLÍVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

CARRERA DE SOFTWARE

TRABAJO DE INTEGRACIÓN CURRICULAR
PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SOFTWARE

FORMA: PROYECTO TECNOLÓGICO

TEMA:

SISTEMA WEB INTEGRAL PARA LA GESTIÓN ADMINISTRATIVA, FINANCIERA Y DE INCIDENCIAS EN INMUEBLES DE ALQUILER PARA ESTUDIANTES DE LA UNIVERSIDAD ESTATAL DE BOLÍVAR, PARROQUIA GUANUJO, 2026.

AUTOR:

KEVIN ANDRES ANDRADE PACHERRIS

DIRECTOR:

Dr. HENRY FERNANDO VALLEJO BALLESTEROS

GUARANDA – ECUADOR

2026

TEMA

SISTEMA WEB INTEGRAL PARA LA GESTIÓN ADMINISTRATIVA, FINANCIERA Y DE INCIDENCIAS EN INMUEBLES DE ALQUILER PARA ESTUDIANTES DE LA UNIVERSIDAD ESTATAL DE BOLÍVAR, PARROQUIA GUANUJO, 2026.

DEDICATORIA

Dedico esta tesis primeramente a Dios, que a lo largo de todo este proceso me ha llenado de bendiciones, dándome la fuerza y la sabiduría necesarias para seguir adelante.

A mi papá, quien ha sido el pilar fundamental para la culminación de este proyecto; gracias por estar presente en cada instante, por impulsarme a no rendirme y por brindarme su amor incondicional, sus consejos y la inspiración que necesitaba en los momentos más difíciles.

A mis hermanos, por acompañarme en cada etapa de mi vida y por estar ahí, de manera incondicional, cada vez que acudí a su ayuda.

Kevin Andres Andrade Pacherris

AGRADECIMIENTO

Primeramente, agradezco a Dios por regalarme la vida y la salud, por su amor infinito y por guiar cada uno de mis pasos, porque tengo la certeza de que sin su bendición ningún plan llega a realizarse.

A mi papá, por su amor incondicional y su apoyo inquebrantable a lo largo de toda mi vida. Gracias por cada sacrificio realizado para que yo pudiera alcanzar esta meta; a pesar de cualquier circunstancia, siempre estuvo ahí alentándome a seguir adelante y a no rendirme jamás. Este logro es el resultado de su esfuerzo tanto como del mío.

A mis hermanos, Axel, Keneth y Kaylee, por su alegría, su cariño y por ser un apoyo constante en cada momento. Su presencia en mi vida ha sido un motor fundamental para no dejar de esforzarme.

A mi mamá, que a pesar de la distancia, aún valoro profundamente los consejos y enseñanzas que me brindó durante mi infancia. A mi novia, que llegó en el momento justo para darme el ánimo y apoyo que necesitaba en esta recta final.

A la Universidad Estatal de Bolívar, por abrirme sus puertas, permitirme formar parte de su comunidad y brindarme las herramientas necesarias para mi formación y crecimiento profesional.

A mi director de tesis, el Dr. Henry Vallejo, por su valiosa guía, su paciencia y por compartir sus conocimientos con absoluta generosidad, orientándome con éxito en el desarrollo de este proyecto.

A mis pares académicos, el Ing. Darwin Carrión y el Ing. Edgar Ribadeneira, por su tiempo, sus observaciones constructivas y su criterio profesional, los cuales enriquecieron significativamente este trabajo.

Kevin Andres Andrade Pacherris

CERTIFICADO DE VALIDACIÓN

UEB
UNIVERSIDAD ESTATAL DE BOLIVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

CERTIFICADO DE VALIDACIÓN

Dr. Henry Vallejo, Ing. Darwin Carrión e Ing. Edgar Ribadeneira, en su orden Director y Pares Académicos del Trabajo de Integración Curricular "SISTEMA WEB INTEGRAL PARA LA GESTIÓN ADMINISTRATIVA, FINANCIERA Y DE INCIDENCIAS EN INMUEBLES DE ALQUILER PARA ESTUDIANTES DE LA UNIVERSIDAD ESTATAL DE BOLÍVAR, PARROQUIA GUANUJO, 2026." desarrollado por el señor Kevin Andres Andrade Pacherris.

CERTIFICAN

Que, luego de revisado el Trabajo de Integración Curricular en su totalidad, cumple con las exigencias académicas de la carrera SOFTWARE, por lo tanto, autorizamos su presentación y defensa.

Guaranda, 02 del Julio del 2026

Dr. Henry Vallejo
Director

Ing. Darwin Carrión
Par Académico

Ing. Edgar Ribadeneira
Par Académico

DERECHOS DE AUTOR

UEB UNIVERSIDAD ESTATAL DEBOLIVAR

BIBLIOTECA GENERAL

DERECHOS DE AUTOR

Yo **Kevin Andres Andrade Pacherris** portador de la Cédula de Identidad No **1250181607** en calidad de autor y titular de los derechos morales y patrimoniales del Trabajo de Titulación:
SISTEMA WEB INTEGRAL PARA LA GESTIÓN ADMINISTRATIVA, FINANCIERA Y DE INCIDENCIAS EN INMUEBLES DE ALQUILER PARA ESTUDIANTES DE LA UNIVERSIDAD ESTATAL DE BOLÍVAR, PARROQUIA GUANUJO, 2026, modalidad **Proyecto Tecnológico**, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN, concedemos a favor de la Universidad Estatal de Bolívar, una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservo a mi favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo a la Universidad Estatal de Bolívar, para que realice la digitalización y publicación de este trabajo de titulación en el Repositorio Digital, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

El autor declara que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.



Kevin Andres Andrade Pacherris

INDICE DE CONTENIDOS

DEDICATORIA iii

AGRADECIMIENTO iv

CERTIFICADO DE VALIDACIÓN v

DERECHOS DE AUTOR vi

INDICE DE CONTENIDOS vii

INDICE DE TABLAS xiii

INDICE DE FIGURAS xvi

INTRODUCCIÓN 19

RESUMEN 20

ABSTRACT 21

KICHUA 22

CAPÍTULO I 23

FORMULACIÓN GENERAL DEL PROYECTO 23

1.1 Tema 23

1.2 Descripción del Problema 23

1.3 Justificación 24

1.4 Objetivos: General y Específicos 25

1.4.1 General 25

1.4.2 Específicos 25

2 CAPÍTULO II 26

MARCO TEÓRICO....26
2.1 Antecedentes....26
2.2 Científico....30
2.2.1 Metodología de desarrollo de software....30
2.2.2 Metodologías Agiles....30
2.2.3 Alquiler de bienes inmuebles....34
2.3 Conceptual....35
Alfabetización digital....35
API....35
Arquitectura....35
Aseguramiento de la Calidad del Software (SQA)....35
Backend....36
Ciclo de vida de desarrollo de software....36
CRUD....36
ERS....36
Framework....36
Frontend....37
HTTP / HTTPS....37
Inmuebles....37
Laravel....37
PHP....38

PostgreSQL 38

Scrum 38

UI/UX 38

XP (Extreme Programming) 38

2.4 Legal 39

2.4.1 Ley de Inquilinato 39

2.4.2 Ley Orgánica de Protección de Datos Personales 40

2.4.3 Código Orgánico de Economía Social de los Conocimientos (INGENIOS) 41

2.5 Georreferencial 42

3 CAPÍTULO III 43

METODOLOGÍA 43

3.1 Metodología de Desarrollo de Software 43

3.1.1 Adaptación de XP para un desarrollador individual 43

3.1.2 Faces de la Metodología Extreme Programming XP 44

Fase I - Planificación del proyecto 44

Fase II - Diseño 45

Fase III - Codificación 45

Fase IV - Pruebas 45

3.2 Técnicas e Instrumentos de Recopilación de Datos 46

3.2.1 Entrevista 46

3.2.2 Encuesta 47

4 CAPITULO IV 48

INGENIERÍA DEL PROYECTO 48

4.1 Fase I: Planificación 48

4.1.1 Universo 48

4.1.2 Procesamiento de datos 49

4.1.2.1 Análisis de Datos 49

4.1.2.3 Análisis de las Encuestas 67

Fragilidad Administrativa y Movilidad Estudiantil 67

Brecha en la Gestión Financiera 68

Eficiencia Operativa e Incidencias 68

4.1.3 Historias de usuarios 68

4.1.4 Requerimientos no Funcionales 81

4.1.5 Plan de Entregas 83

4.2 Fase II: Diseño 85

4.2.1 Modelo de Vistas Arquitectónicas 85

Vista de Escenarios 87

Vista de Desarrollo 90

Vista Lógica 92

Vista de Procesos 94

Vista de Física 100

4.2.2 Patrón Arquitectónico 101

Modelo 101

Modelo Entidad Relación de la Base de datos 102

Vista 102

Diseño de la Interfaz 103

Controlador 110

Algoritmo de autenticación (Login) y enrutamiento de roles 110

Algoritmo de vinculación de múltiples partes 111

Algoritmo de filtrado dinámico y seguridad 112

Proxy transaccional de archivos nube 113

4.3 Fase III: Codificación 114

Capa de presentación (Frontend) 115

Capa de lógica de negocio (Backend) 115

Capa de datos (PostgreSQL) 116

Infraestructura y configuración de la base de datos (PostgreSQL) 116

Configuración de variables de entorno 117

Estructura de contratos y vinculaciones 117

Organización de rutas de la API (web.php) 117

4.4 Pruebas 118

Pruebas unitarias 119

AuthContext 119

ProtectedRoute 119

Componente RegisterPage 120

Pruebas de integración 122

RegistroInmueblePage 122

VincularPage 124

Pruebas de mantenibilidad y seguridad 126

Análisis crítico y arquitectura de la seguridad transaccional 129

1. Autenticación robusta y gestión de información mínima 129

2. Cifrado criptográfico de la persistencia 130

3. Validación estricta de identidad y unicidad en casos de uso 130

4. Control de acceso basado en roles (RBAC) 130

5. Gestión de archivos binarios aislados 131

Validación de capas de seguridad en la API 131

5 CONCLUSIONES 134

6 RECOMENDACIONES 135

7 BIBLIOGRAFÍA 136

8 ANEXOS 140

INDICE DE TABLAS

Tabla 1. Metodologías Agiles....31

Tabla 2. Tipos de bienes inmuebles....34

Tabla 3. Número de participantes de la encuesta 48

Tabla 4. Tabulación de datos de la pregunta 1 a arrendatarios....49

Tabla 5. Tabulación de datos de la pregunta 2 a arrendatarios....50

Tabla 6. Tabulación de datos de la pregunta 3 a arrendatarios....51

Tabla 7. Tabulación de datos de la pregunta 4 a arrendatarios....52

Tabla 8. Tabulación de datos de la pregunta 5 a arrendatarios....53

Tabla 9. Tabulación de datos de la pregunta 6 a arrendatarios....54

Tabla 10. Tabulación de datos de la pregunta 7 a arrendatarios 55

Tabla 11. Tabulación de datos de la pregunta 8 a arrendatarios 56

Tabla 12. Tabulación de datos de la pregunta 9 a arrendatarios 57

Tabla 13. Tabulación de datos de la pregunta 10 a arrendatarios 58

Tabla 14. Tabulación de datos de la pregunta 11 a arrendatarios 59

Tabla 15. Tabulación de datos de la pregunta 1 a arrendadores 60

Tabla 16. Tabulación de datos de la pregunta 2 a arrendadores 61

Tabla 17. Tabulación de datos de la pregunta 3 a arrendadores 62

Tabla 18. Tabulación de datos de la pregunta 4 a arrendadores 63

Tabla 19. Tabulación de datos de la pregunta 5 a arrendadores 64

Tabla 20. Tabulación de datos de la pregunta 6 a arrendadores 65

Tabla 21. Tabulación de datos de la pregunta 7 a arrendadores 66

Tabla 22. Historia de Usuario Inicio de sesión y recuperación de contraseña 68

Tabla 23. Historia de Usuario Registro de Usuario 69

Tabla 24. Historia de Usuario Registro de Inmueble 71

Tabla 25. Historia de Usuario Vinculación de Usuarios con el Inmueble 71

Tabla 26. Historia de Panel de Auditoría y Validación de Cobros 72

Tabla 27. Historia de Panel de Pagos y Justificativos Digitales 73

Tabla 28. Historia de Gestión de Contratos y Normas de Convivencia 74

Tabla 29. Historia de Control de Gastos 75

Tabla 30. Historia de Seguimiento de Cobros 75

Tabla 31. Historia de Usuario Seguimiento de pagos 76

Tabla 32. Historia de Gestión de Incidencias mediante Tickets 77

Tabla 33. Historia de Registro de Pagos con Comprobante Digital 78

Tabla 34. Historia de Supervisión de Estado Financiero y Residencial 79

Tabla 35. Historia de Generación de Reportes PDF 79

Tabla 36. Historia de Configuración de Seguridad y Accesos 80

Tabla 37. Requerimientos no funcionales 81

Tabla 38. Tabla para la estimación de entregas por cada historia de usuario 84

Tabla 39. El Modelo de “4+1” Vistas de la Arquitectura del Software 86

Tabla 40. Tabla de colores para el sistema GestInWEB 103

Tabla 41. Estructura de Codificación 114

Tabla 42. Rutas api....118

Tabla 43. Pruebas Unitarias con Vitest....120

Tabla 44. Prueba de integración a la interfaz RegistroInmueblePage....122

Tabla 45. Prueba de integración a la interfaz VincularPage....124

Tabla 46. Métricas e indicadores arrojados del escaneo....128

Tabla 47. Validación de Capas de Seguridad en la API....131

Tabla 48. Presupuesto Ejecutado....144

INDICE DE FIGURAS

Figura 1. Fases de la Metodología XP 46

Figura 2. Diagrama de la tabulación de datos de la pregunta 1 a arrendatarios.....50

Figura 3. Diagrama de la tabulación de datos de la pregunta 2 a arrendatarios.....51

Figura 4. Diagrama de la tabulación de datos de la pregunta 3 a arrendatarios.....52

Figura 5. Diagrama de la tabulación de datos de la pregunta 4 a arrendatarios.....53

Figura 6. Diagrama de la tabulación de datos de la pregunta 5 a arrendatarios.....53

Figura 7. Diagrama de la tabulación de datos de la pregunta 6 a arrendatarios.....55

Figura 8. Diagrama de la tabulación de datos de la pregunta 7 a arrendatarios.....56

Figura 9. Diagrama de la tabulación de datos de la pregunta 8 a arrendatarios.....57

Figura 10. Diagrama de la tabulación de datos de la pregunta 9 a arrendatarios.....58

Figura 11. Diagrama de la tabulación de datos de la pregunta 10 a arrendatarios.....59

Figura 12. Diagrama de la tabulación de datos de la pregunta 11 a arrendatarios.....60

Figura 13. Diagrama de la tabulación de datos de la pregunta 1 a arrendadores 61

Figura 14. Diagrama de la tabulación de datos de la pregunta 2 a arrendadores 61

Figura 15. Diagrama de la tabulación de datos de la pregunta 3 a arrendadores 62

Figura 16. Diagrama de la tabulación de datos de la pregunta 4 a arrendadores 64

Figura 17. Diagrama de la tabulación de datos de la pregunta 5 a arrendadores 65

Figura 18. Diagrama de la tabulación de datos de la pregunta 6 a arrendadores 66

Figura 19. Diagrama de la tabulación de datos de la pregunta 7 a arrendadores 67

Figura 21. Diagrama de casos de usos del módulo del Arrendatario 88

Figura 22. Diagrama de casos de usos del modulo arrendador 89

Figura 23. Diagrama de casos de usos del modulo de control parental 90

Figura 24. Diagrama de Paquetes del sistema 91

Figura 25. Diagrama de Clases del sistema 93

Figura 26. Diagrama de secuencia de Inicio de sesión 95

Figura 27. Diagrama de secuencia del Registro de inmuebles 96

Figura 28. Diagrama de secuencia de Vinculación de inmuebles 97

Figura 29. Diagrama de secuencia de Pagos y cobros 98

Figura 30. Diagrama de secuencia de Generar reportes 99

Figura 31. Diagrama de secuencia de las Incidencias 100

Figura 32. Diagrama de despliegue del sistema 101

Figura 33. Diagrama Entidad-Relación del sistema 102

Figura 34. Logo del sistema 103

Figura 35. Interfaz de Inicio de sesión 104

Figura 36. Interfaz de registro 105

Figura 37. Interfaz de registro de inmueble 106

Figura 38. Interfaz de vinculación de usuarios con el inmueble 107

Figura 39. Interfaz de control de pagos 108

Figura 40. Interfaz de registro de pagos 109

Figura 41. Interfaz de generación de reportes 110

Figura 42. Archivo de propiedades de sonarQube 126

Figura 43. Comandos para inciar el escaneo 127

Figura 44. Vista de resultados del escaneo 127

Figura 45. Cronograma (Gantt) 142

Figura 46. Revisión de documento y requisitos del sistema - Ing. Edgar Ribadeneira 159

Figura 47. Revisión de documento y requisitos del sistema – Ing Darwin Carrión 159

Figura 48. Revisión del sistema y sus funcionalidades – Ing. Darwin Carrión 160

Figura 49. Revisión del sistema y sus funcionalidades – Ing. Edgar Ribadeneira 160

Figura 50. Revisión del sistema y sus funcionalidades 1 - Arrendador 160

Figura 51. Revisión del sistema y sus funcionalidades 2 - Arrendador 161

Figura 52. Revisión del sistema y sus funcionalidades 3 - Arrendador 161

Figura 53. Revisión del sistema y sus funcionalidades 4 - Arrendador 162

Figura 54. Revisión del sistema y sus funcionalidades 5 - Arrendador 162

Figura 55. Revisión del sistema y sus funcionalidades 6 - Arrendador 162

INTRODUCCIÓN

El crecimiento demográfico y la expansión académica de la Universidad Estatal de Bolívar han cambiado considerablemente la dinámica urbana en el municipio de Guanujo. La afluencia de estudiantes de diversas regiones lleva a un aumento en la demanda de vivienda, lo que obliga a lidiar con el mercado inmobiliario. Sin embargo, la infraestructura tradicional no es lo suficientemente eficiente para satisfacer la demanda y, por lo tanto, el mercado está impulsado por un alto número de transacciones y datos (procesos y métodos) que están desactualizados y no se comprenden bien.

Se construyó un sistema web para ayudar en una solución completa que satisfaga las necesidades del proyecto centralizando y automatizando el trabajo administrativo, financiero y operativo de los arrendadores y estudiantes arrendatarios de la Universidad Estatal de Bolívar de la parroquia de Guanujo. Este proyecto fue diseñado con un enfoque muy ágil de la metodología Extreme Programming (XP), ya que utiliza un diseño simple, refactorización continua y desarrollo iterativo para trabajar con un solo desarrollador. La arquitectura del sistema se basa en el patrón de diseño de modelo en capas y el Modelo-Vista-Controlador (MVC), y se muestra arquitectónicamente robusta en el Modelo de Vista 4+1 de Kruchten. Las interfaces dinámicas se construyen en el frontend utilizando React.js y Vite, y están conectadas con un backend estructurado construido con PHP y Laravel. La persistencia de datos transaccionales y el esquema relacional se almacenan únicamente en una base de datos PostgreSQL, y los archivos multimedia, recibos y documentos PDF se almacenan solo en buckets de Supabase.

La estructura general de este documento describe cómo se realiza el trabajo. El Capítulo I define el problema, justifica la investigación y pretende alcanzar los objetivos; el Capítulo II desarrolla el marco conceptual visto en los antecedentes del estudio y fundamentos académicos, conceptuales, legales y otros fundamentos legales para el proyecto y el Capítulo III introduce el enfoque ágil adoptado para la recolección de datos. El Capítulo IV es la última parte, que describe el proceso de ingeniería para el proyecto que incluye el procesamiento estadístico del universo muestral, diseño de diagramas, codificación arquitectónica y pruebas de calidad del software.

RESUMEN

Este proyecto tecnológico aborda la informalidad y la gestión manual en el alquiler de propiedades destinadas a estudiantes de la Universidad Estatal Bolívar en la parroquia de Guanujo. El diagnóstico de campo, realizado con un universo de 50 inquilinos y 25 propietarios, reveló que el 64% de las estancias residenciales se gestionan mediante acuerdos verbales y que el 72% de los propietarios se basan exclusivamente en registros físicos. Esta situación genera vulnerabilidad legal, inestabilidad financiera por transacciones en efectivo sin comprobantes de seguimiento y retrasos operativos para la resolución de daños a la infraestructura. Para superar estas deficiencias, se desarrolló e implementó el sistema web integral GestInWEB, diseñado para centralizar y automatizar los flujos de trabajo operativos, financieros y administrativos del proceso de alquiler.

La plataforma se construyó utilizando la metodología ágil Extreme Programming (XP), adaptada para desarrolladores individuales, organizando el ciclo de vida del software mediante iteraciones cortas y aplicando el modelo de vista arquitectónica 4+1 junto con el patrón estructural Modelo-Vista-Controlador (MVC). El ecosistema tecnológico emplea React.js y Vite para la capa de presentación, PHP con el framework Laravel para la lógica de negocio y PostgreSQL para la persistencia relacional estructurada. Se integró Supabase externamente y exclusivamente para el almacenamiento en la nube de archivos binarios, documentos multimedia e imágenes de soporte, manteniendo los datos pesados aislados para optimizar el rendimiento del servidor principal.

El sistema incluye módulos específicos para el registro detallado de propiedades, la formalización de contratos con múltiples partes, un panel de auditoría financiera para validar capturas de pantalla de transferencias bancarias, alertas visuales por pagos atrasados y un canal de gestión de incidentes que utiliza tickets con soporte fotográfico. Además, se implementó un rol de control parental diferenciado, que permite a los representantes legales monitorear la situación financiera y de vivienda del estudiante con permisos de visualización restringidos. Las fases de prueba confirman que el sistema profesionaliza el sector local de alojamiento estudiantil, garantizando la honestidad en las transacciones y la seguridad de la información sensible de acuerdo con la legislación ecuatoriana vigente.

ABSTRACT

This technology project addresses informality and manual management in the rental of properties intended for students of the State University of Bolívar in the Guanujo parish. The field diagnosis, conducted with a sample of 50 tenants and 25 landlords, revealed that 64% of residential stays are managed through verbal agreements and that 72% of landlords rely exclusively on records in physical ledgers. This situation creates legal vulnerability, financial instability from cash transactions without traceable receipts, and operational delays in addressing infrastructure damage. To overcome these shortcomings, the comprehensive web-based system GestInWEB was developed and implemented, designed to centralize and automate the operational, financial, and administrative workflows of the rental process.

The platform was built using the agile Extreme Programming (XP) methodology, adapted for individual developers, organizing the software lifecycle through short iterations and applying the 4+1 architectural view model along with the Model-View-Controller (MVC) structural pattern. The technology ecosystem employs React.js and Vite for the presentation layer, PHP with the Laravel framework for business logic, and PostgreSQL for structured relational persistence. Supabase was integrated externally and exclusively for cloud storage of binary files, multimedia documents, and supporting images, keeping heavy data isolated to optimize the main server's performance.

The system includes specific modules for detailed property registration, formalizing contracts involving multiple parties, a financial audit dashboard to validate screenshots of bank transfers, visual alerts for late payments, and an incident management channel that uses tickets with photographic support. Furthermore, a differentiated parental control role was implemented, allowing legal representatives to monitor the student's financial and housing status with restricted viewing permissions. The testing phases confirm that the system professionalizes the local student housing sector, guaranteeing transactional honesty and the security of sensitive information in accordance with current Ecuadorian law.

KICHUA

Kay antanikik rurayka Guanujo kitillipi, Bolívar Hatun Yachay Wasipi yachakukkunapak wasikunata mañachiypi, mana kamachiypi kashkata, makiwanlla ruray llakikunatapashmi allichin. 50 mañakkunata, 25 wasiyukkunata tapushkamantaka, 64% patsakrikpi shimiwanlla ari nishkawan purishkata rikuchirka, shinallatak 72% patsakrikpi wasiyukkuna makiwanlla killkashka pankakunapi charishkatapashmi rikuchirka. Kayka llakikunatami apamun: kamachiykunapi mana wankurishka, kullki mañachiypi panka illak cashkamanta mana alli yupaykuna, shinallatak wasipi imapash wakllikpi mana utka allichishkapash.

Kay llakikunata sapimanta allichinkapakmi, GestInWEB shutiyuk llikata rurarka. Kay llikaka wasi mañachiypi ruraykunata, kullkikunata, pushay ruraykunatapash tantachinkapak, antanikikllapi rurachunpashmi wallpashka karka.

Kay llikata rurankapakka Extreme Programming (XP) nishka rurayñanwanmi llankarka, shuklla rurakpak allichishka, antanikikpa kawsayta utka kutin ruraykunawan allichishpa, shinallatak 4+1 rikuykunata, Modelo-Vista-Controlador (MVC) nishkatapash churashpa. Llikapika React.js, Vite nishkakunatami rikuchinkapak churarka, PHP shinallatak Laravel framework nishkata ruraykunapak, PostgreSQL willaykunata alli wakaychinkapakpash. Supabase llikataka hawa puyupi llasa willaykunata, shuyukunata, rikuchik killkakunata wakaychinkapakmi sapalla churarka, hatun llikapa rurayta utkayachinkapak.

Kay llikaka wasikunata alli killkankapak, ashtawan runakunawan rimanakuypi killkankapak, kullki kachashkakunata rikunkapak pankata, kullki manu kakpi rikuchikkunata, shinallatak wakllishkakunata shuyuwan rikuchishpa allichinkapakpashmi ruraykunata charin. Shinallatak, yaya mamakunapak shuk rikuytami rurarka, paykunapa wawakuna mashna kullkita minishtin, wasipi imashina kawsan rikuchunlla, mana imatapash yapa rikuchun. Llikata llankachishpaka, kay antanikik ruray wasi mañachiyta alliyachishkatami rikuchin, tukuy ruraykunata chuyata rikuchishpa, shinallatak Ecuador mama llaktapa kamachiykunapi nishkashina willaykunata alli wakaychishpapash.

CAPÍTULO I

FORMULACIÓN GENERAL DEL PROYECTO

1.1 Tema

Sistema web integral para la gestión administrativa, financiera y de incidencias en inmuebles de alquiler para estudiantes de la Universidad Estatal de Bolívar, parroquia Guanujo, 2026.

1.2 Descripción del Problema

En la parroquia Guanujo, el crecimiento de la población universitaria ha generado una alta demanda de bienes inmuebles destinados al arrendamiento. Actualmente, la gestión de estos alquileres se realiza de manera empírica y manual, utilizando cuadernos u hojas de cálculo básicas; inclusive, en muchos casos no existe un registro formal, limitándose a acuerdos verbales carentes de respaldo técnico y legal. Esta ausencia de formalidad sumada a métodos de registro obsoletos, genera una alta vulnerabilidad ante errores humanos como la pérdida de los datos de los arrendatarios y la carencia de evidencia física, lo cual dificulta el control preciso de los contratos y la administración eficiente de las unidades habitacionales.

En el aspecto financiero se presentan complicaciones críticas, dado a que no existe un sistema automatizado para el seguimiento y control de los pagos mensuales, servicios básicos o depósitos de garantía por inmueble. Los arrendatarios suelen olvidar sus fechas de pago y los arrendadores carecen de herramientas para generar comprobantes de inmediato. Esta falta de transparencia financiera genera fricciones constantes entre ambas partes, afectando la confianza y la organización económica de los estudiantes que dependen de presupuestos limitados.

La gestión de incidencias y mantenimientos es prácticamente inexistente de forma estructurada; cuando ocurre un daño en el inmueble o surge un conflicto de convivencia, el reporte se realiza mediante llamadas o mensajes de texto que son olvidados por el arrendador, esto hace que la falta de un registro histórico de daños o quejas impide planificar mantenimientos preventivos, lo que acelera el deterioro el inmueble y disminuye la calidad de vida de los estudiantes residentes.

Sin las herramientas digitales en la parroquia que ayudarían con este problema, los propietarios y los inquilinos estudiantes tienen que hacer el arduo trabajo de gestionar su alquiler mediante

registros manuales, acuerdos verbales y comunicación informal. Esta falta de formalización dificulta que los propietarios gestionen sus propiedades de manera profesional y eficiente, pero también impide que los inquilinos tengan la documentación que muestra todos sus derechos. Por lo tanto, es necesario crear una solución web que conecte todos estos procesos en un solo lugar y sea fácilmente accesible y estable.

1.3 Justificación

La gestión de alquileres en la parroquia de Guanujo es muy práctica e informal; por lo que los problemas tanto para los propietarios como para los inquilinos estudiantes de la Universidad Estatal de Bolívar nunca se han resuelto. Este sistema nace como una respuesta directa a esa realidad.

El principal beneficio del sistema es la centralización de la información, lo que fomenta una relación más transparente y equitativa entre propietario e inquilino. Los estudiantes contarán con una interfaz intuitiva desde la cual podrán consultar sus contratos, registrar pagos mediante comprobantes digitales y reportar incidencias o daños a través de tickets categorizados por urgencia eliminando así los malentendidos frecuentes y mejora las condiciones de habitabilidad.

Para los arrendadores, el sistema es una herramienta real de profesionalización que incluye; la automatización de recordatorios de pago que reduce la morosidad, el registro histórico de transacciones que facilita el control financiero, la gestión oportuna de mantenimientos que evita que pequeños problemas se conviertan en reparaciones costosas a largo plazo, inclusive la generación automática de recibos en PDF y el acceso a reportes financieros detallados que permiten una planificación más sólida y documentada.

Técnicamente, el sistema está basado en una estructura modular que garantiza la integridad de los datos y la eficiencia operativa. El rol de arrendador centraliza la administración de sus inmuebles, la formalización de contratos, el seguimiento de los pagos y el manejo de incidencias, en cambio, el arrendatario puede observar la información del inmueble, gestionar sus pagos e incidencias, y también un componente distintivo del sistema es el módulo de control parental, este módulo permite a padres o representantes legales monitorear indirectamente la situación financiera y de habitabilidad del estudiante, sin tener la capacidad de realizar cambios dentro del sistema.

El rol de control parental responde a una necesidad específica del contexto universitario y hace que este proyecto se diferencie de otras soluciones similares. Esta plataforma busca eliminar la informalidad en la gestión de arriendos de inmuebles para estudiantes en la parroquia Guanujo, estableciendo así un canal de comunicación eficiente, seguro y accesible que beneficia a toda la comunidad involucrada: estudiantes, propietarios e incluso familias.

El desarrollo de este sistema se relaciona con la línea de Ingeniería de Software y la sublínea de Diseño e Implementación de Sistemas de Información.

1.4 Objetivos: General y Específicos

1.4.1 General

Implementar el sistema web integral para la gestión administrativa, financiera y de incidencias en inmuebles de alquiler para estudiantes de la Universidad Estatal de Bolívar, parroquia Guanujo, 2026.

1.4.2 Específicos

- Analizar los requerimientos de los arrendadores y estudiantes de la parroquia Guanujo para la obtención de historias de usuario.
- Diseñar la arquitectura de software, el modelado de la base de datos y la interfaz de usuario (UI/UX) priorizando la usabilidad y la seguridad de la información.
- Desarrollar los módulos de gestión de contratos, control de pagos recurrentes y levantamiento de tickets de incidencias utilizando un stack tecnológico robusto.
- Validar el funcionamiento del sistema a través de pruebas funcionales y de usuario.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes

En las últimas décadas, la transición hacia la transformación digital ha redefinido la gestión administrativa en diversos sectores. La evolución de las herramientas informáticas ha permitido que la automatización de procesos se convierta en el eje central para mejorar la eficiencia operativa.

Este avance es particularmente relevante en sectores que tradicionalmente han dependido de procesos manuales, donde la implementación de software especializado garantiza una mayor precisión y control de la información.

Un hecho relevante que motivó la creación del presente proyecto, titulado “Sistema web integral para la gestión administrativa, financiera y de incidencias en inmuebles de alquiler para estudiantes de la Universidad Estatal de Bolívar, parroquia Guanujo, 2026”, fue la creciente demanda de alojamiento de estudiantes cercano al campus universitario, sumada a la falta de herramientas digitales para administrar contratos, cobros y reportes de incidentes.

En Guanujo, el registro de inmuebles todavía se apoya con métodos manuales: libretas, carpetas físicas y archivos de Excel que no siempre están unificados ni se actualizan con el mismo criterio. Este escenario, aunque común, genera problemas concretos: morosidad que se acumula sin alertas claras, información repetida o contradictoria y tiempos de respuesta largos ante averías reportadas por los estudiantes. Cuando el control se fragmenta entre papeles, mensajes sueltos y archivos dispersos, la administración pierde continuidad y el estudiante lo percibe en su vida cotidiana. Al mismo tiempo, el arrendador asume costos y vacíos de seguimiento que afectan su ingreso y su capacidad de planificación. Desde esta realidad surge la demanda de un sistema web accesible, que concentre datos, eventos y documentación en un solo entorno.

En Guanujo, el registro de alquiler de propiedades todavía depende de métodos manuales: cuadernos, carpetas físicas y archivos de Excel que no siempre están consolidados o actualizados con los mismos criterios. Esta es una situación común, pero hay problemas; la morosidad se acumula sin alertas oportunas, información repetida o contradictoria y tiempos

de respuesta lentos ante las averías que experimentan los estudiantes. Es en este contexto que, sin una gestión consistente la administración pierde continuidad; y cuando el estudiante lo experimenta en su vida diaria, el propietario incurre en costos y brechas en el monitoreo que afectan sus ingresos y capacidad de planificación. Por eso hay una necesidad de un sistema web que pueda tener todas estas cosas en un solo lugar, toda la información pueda ser recopilada y todo tipo de cosas estén organizadas.

Martínez (2018), en el trabajo “Sistema para la Gestión de los inmuebles”, presentado en el VIII Congreso Iberoamericano de Ingeniería de Proyectos en La Habana, Cuba, propone automatizar los procesos de gestión inmobiliaria para mejorar la toma de decisiones estratégicas y la eficiencia en el control de activos. El propósito del proyecto era desarrollar una herramienta informática para gestionar archivos digitales, criterios de clasificación y estado de conservación, con un enfoque en los registros manuales y la falta de una base de datos central.

Metodológicamente basado en RUP (Proceso Unificado Racional) y UML, el sistema se organizó según casos de uso y requisitos. En términos de tecnología, se utilizaron PHP (framework Symfony) y PostgreSQL como gestor de base de datos (con integridad transaccional y seguridad de la información).

Encontraron que los archivos digitalizados y el monitoreo continuo del estado físico de las propiedades hacen que el mantenimiento preventivo y correctivo sea más anticipado y controlado. En esta tesis tenemos dos aspectos: apoyamos el uso de PHP y PostgreSQL como soporte tecnológico y justificamos la adición de módulos para el control del estado de las propiedades y la gestión de documentos digitales que están en línea con el sistema existente en la gestión documental.

Barahona Blanco (2020), en su trabajo de fin de grado titulado "Diseño y Desarrollo de una Aplicación Web para la Gestión de Inmuebles en Alquiler y el Control de Pagos", presentado en la Universidad Politécnica de Madrid, propone una solución integral al desafío administrativo que enfrentan los propietarios. Fue el estudio de la creación de una plataforma para controlar los gastos de la propiedad y para el adecuado seguimiento de todos los ingresos que los inquilinos están generando.

Se utilizó el lenguaje de programación Python en el desarrollo de la aplicación web junto con el framework Django que se utilizó para gestionar consultas de base de datos para conectar inquilinos, propiedades y registros de alquiler.

Los hallazgos de este informe destacan que automatizar el control de pagos no solo reduce la presión administrativa del propietario, sino que también proporciona transparencia al inquilino para llevar un registro de sus pagos. Este antecedente fue fundamental para el presente proyecto, validando técnicamente la necesidad de un módulo de pagos y generación de recibos de esta manera refuerza la importancia de la generación de comprobantes para evitar conflictos por falta de supervisión en los ingresos de los arrendadores y a su vez los arrendatarios tienen un mayor control de sus pagos.

Reinoso Alvarado y Revelo Herrera (2022), en su trabajo de titulación titulado "Implementación de un sistema para la gestión de inmuebles en el barrio Granda Centeno de la ciudad de Quito", proponen una solución tecnológica para centralizar y optimizar la administración de propiedades. El estudio tuvo como objetivo principal desarrollar una herramienta que permitiera gestionar de forma eficiente la información de inmuebles destinados a la renta y venta, mitigando los errores derivados del manejo manual de datos.

Ellos emplearon la metodología ágil SCRUM, estructurando el desarrollo en ciclos iterativos que permitieron la validación constante de requerimientos, también se utilizó el lenguaje Python con el framework Django para el backend y PostgreSQL como su gestor de bases de datos, garantizando la integridad y escalabilidad de la información.

Los resultados del estudio evidenciaron una optimización notable en los procesos operativos; al unificar los flujos de información, se eliminó la redundancia de registros y se facilitó la emisión de reportes administrativos exactos. Esta investigación es de vital relevancia para el presente proyecto, ya que, no solo refuerza la viabilidad técnica de utilizar PostgreSQL para la persistencia de datos inmobiliarios, sino que también intensifica la necesidad de implementar interfaces intuitivas para profesionalizar la gestión de alquileres, sirviendo como un referente comparativo directo para el sistema web orientado a la comunidad estudiantil de la Universidad Estatal de Bolívar.

El problema de gestionar alojamientos universitarios no es nuevo, pero las soluciones digitales que lo abordan con rigor técnico siguen siendo escasas. Muyudumbay Alvarez (2022) se

propuso cambiar eso. En su trabajo de titulación "Desarrollo de una aplicación web de gestión inmobiliaria de alquiler estudiantil en la ciudad de Riobamba", desarrollado en la Escuela Superior Politécnica de Chimborazo, diseñó una plataforma que centraliza la oferta habitacional y conecta directamente a arrendadores con estudiantes. Los resultados hablaron solos: la consolidación de datos redujo la redundancia, los reportes administrativos ganaron precisión, y quizás lo más significativo los usuarios describieron la experiencia como adecuada y placentera. La consistencia visual y la operación clara del sistema no fueron casualidad; fueron el producto de decisiones de diseño deliberadas.

Para el presente proyecto, este antecedente no es solo una referencia más, es una validación. trabajar con estudiantes universitarios implica atender a un público con expectativas altas respecto a la experiencia digital y poca tolerancia a sistemas confusos; que una interfaz bien estructurada sea determinante para la adopción del sistema ya fue demostrado no hay que especularlo.

A eso se suma la experiencia de Pacheco Astudillo y Sangoquiza Iza (2024), quienes en "Desarrollo de una aplicación web y móvil para la gestión de inmuebles en el conjunto habitacional Los Arupos de Latacunga" enfrentaron un escenario diferente: la administración de un conjunto habitacional con procesos manuales que urgían ser reemplazados. Su solución integró una plataforma web y una aplicación móvil; los usuarios pudieron registrar propiedades, programar citas y organizar eventos comunitarios en cualquier dispositivo. Este compromiso con la accesibilidad multiplataforma dice algo importante sobre cómo deben diseñarse los sistemas inmobiliarios actuales. Juntos, ambos trabajos confirman tanto la utilidad técnica de PostgreSQL para la persistencia de datos como la necesidad de crear interfaces que los usuarios reales quieran y puedan usar desde el principio; tomamos decisiones sobre el sistema web para la comunidad estudiantil en la Universidad Estatal de Bolívar.

Esta investigación resulta relevante para el desarrollo del sistema en la parroquia Guanujo, pues aparte de que valida la efectividad de las soluciones multiplataforma para el control administrativo de inmuebles, también proporciona un marco comparativo para el diseño de módulos de interacción y gestión de los inmuebles.

Basado en los antecedentes revisados, se determina que es adecuado implementar un "Sistema web integral para la gestión administrativa, financiera y de incidencias en inmuebles de alquiler para estudiantes de la Universidad Estatal de Bolívar, parroquia Guanujo, 2026", dado que la

evidencia académica consultada demuestra de forma consistente que la digitalización de los procesos de gestión inmobiliaria mejora la eficiencia administrativa, reduce los errores derivados del registro manual, optimiza el control financiero y garantiza transparencia en la relación entre arrendadores y arrendatarios.

La incorporación de herramientas tecnológicas como arquitecturas web robustas, gestores de bases de datos relacionales y metodologías ágiles de desarrollo validadas en todos los antecedentes analizados, permite profesionalizar un sector que como en la parroquia de Guanujo aún opera de manera informal. Esto no solo protege el patrimonio del propietario, sino que también garantiza condiciones más justas y documentadas para los estudiantes universitarios que dependen de estos servicios habitacionales.

Estos antecedentes evidencian que la propuesta de un sistema funcional de esta naturaleza es pertinente y técnicamente factible, pues responde a una necesidad real y documentada tanto a nivel local como regional, y contribuye directamente a mejorar la calidad de vida de la comunidad estudiantil de la Universidad Estatal de Bolívar.

2.2 Científico

2.2.1 Metodología de desarrollo de software

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo (Maida & Pacienzia, 2015).

Las metodologías de desarrollo de software son sistemas organizativos que establecen un conjunto de procedimientos, técnicas y recursos para dirigir la elaboración de software. Estas metodologías definen un ciclo de vida del software, que se compone de fases claramente delimitadas, como la planificación, el diseño, la implementación, las pruebas y el mantenimiento.

2.2.2 Metodologías Agiles

“Las metodologías ágiles tienen dos diferencias fundamentales con las metodologías tradicionales; la primera es que los métodos ágiles son adaptativos, no predictivos. La segunda diferencia es que las metodologías ágiles son orientadas a las personas, no orientadas a los

procesos. Las metodologías ágiles son adaptativas. Este hecho es de gran importancia, ya que contrasta con la predictibilidad buscada por las metodologías tradicionales. Con el enfoque de las metodologías ágiles, los cambios son eventos esperados que generan valor para el cliente" (Navarro Cadavid et al., 2013).

Los equipos trabajan con ciclos breves y continuos, llamados iteraciones, donde el software evoluciona constantemente. Esto les da la flexibilidad de ajustarse rápidamente cuando los clientes cambian de opinión o cuando el mercado da un giro inesperado. Al final, esta forma de trabajar reduce el peligro de invertir meses en crear algo que, para cuando esté listo, ya nadie necesite o quiera.

Tabla 1. Metodologías Agiles

Metodología	**XP**	**Scrum**	**Kanban**
Características	Lleva de 5 a 10 iteraciones. Se define el alcance del proyecto mediante la redacción de "historias de usuario" del cliente. Se tienen reuniones diarias de seguimiento para comunicación y compartir problemas y soluciones. El cliente ayuda al desarrollo del proyecto.	Los proyectos se ejecutan en bloques temporales (iteraciones-sprints) de un mes natural (pueden ser de 2 o 3 semanas si así se necesita).	Es un sistema de información que controlade modo armónico la fabricación de los productos necesarios en la cantidad y tiempo necesarios en cada uno de los procesos que tienen lugar tanto en el interior de la fábrica, como entre distintas empresas.

	Programación en parejas.		
Procesos	La programación del software se hace siempre en pareja, lo que se llama programara dos manos. Se asegura con este método que al menos un programador conoce y controla la labor de otro y queda revisado. La ventaja es que se produce mejor código que en base a un programador, aunque la dificultad de la misma sea mayor.	En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión). El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto.	Existe una serie de principios básicos con el fin de obtener el máximo rendimiento de su flujo de trabajo. Visualice lo que hace (su flujo de trabajo). Limite la cantidad de Trabajo en Proceso (límites del TEP). Realice un seguimiento de su tiempo Lectura fácil de indicadores visuales. Identifique los cuellos de botella y elimine lo que resulta descartable.
Roles	Los equipos de un proyecto de esta tipología y	**Cliente:** responsable de	Service request ManagerService Delivery

	magnitud tienen normalmente las siguientes figuras y roles: Clientes, Programadores, Testers, Manager.	conducir el proyecto. **Programador:** Estiman el tiempo y transfieren las historias de usuario a código. **Tester:** encargado de las pruebas. Encargado de seguimiento. **(Tracker):** Hace el seguimiento de acuerdo con la planificación, interés en la velocidad. **Entrenador (Coach):** Guiar y orientar al equipo. Que comprenda las directrices xp. **Gestor (big boss):** Gerente del proyecto.	ManagerReuniones en KanbanReplenishment MeetingDaily KanbanService delivery meetingDelivery Planning meetingStrategy ReviewOperations ReviewRisk Review

Fuente: Armijos Gadvay y Fernández Ninabanda (2023).

2.2.3 Alquiler de bienes inmuebles

La clave está en que no se transfiere la propiedad, sino solo la posesión y el derecho a usar el bien por un tiempo pactado. Al finalizar el plazo, el bien debe devolverse en el estado en que se recibió, salvo el deterioro normal por su uso legítimo.

El artículo 1856 la codificación al Código Civil, dispone: “Arrendamiento es un contrato en que las dos partes se obligan recíprocamente, la una a conceder el goce de una cosa, o a ejecutar una obra o prestar un servicio, y la otra a pagar por este goce, obra o servicio un precio determinado, salvo lo que disponen las leyes del trabajo y otras especiales” (Asamblea Nacional del Ecuador, 2022, art. 1856).

El artículo 1865 la codificación al Código Civil, dispone “El Arrendador está obligado: 1. A entregar al arrendatario la cosa arrendada; 2. A mantenerla en estado de servir para el fin a que ha sido arrendada; y, 3. A librar al arrendatario de toda turbación o embarazo en el goce de la cosa arrendada” (Asamblea Nacional del Ecuador, 2022, art. 1856).

Tabla 2. Tipos de bienes inmuebles

Categoría	Tipo de Inmueble	Descripción
Residencial	Vivienda Unifamiliar	Construcción independiente destinada al núcleo familiar.
Residencial	Departamento	Unidad habitacional dentro de un edificio o conjunto multifamiliar.
Residencial	Habitaciones	Espacios privados dentro de una vivienda con áreas comunes compartidas.
Comercial	Local Comercial	Espacios destinados a la venta de bienes o servicios.
Comercial	Oficinas	Espacios diseñados para labores administrativas o profesionales.
Industrial	Bodegas / Galpones	Espacios de gran tamaño para almacenamiento o producción.

Rural	Terrenos / Quintas	Espacios de tierra destinados a la agricultura o recreación.

Nota. Elaborado por: Andrade P. Kevin.

2.3 Conceptual

Alfabetización digital

La alfabetización digital se enfoca en el desarrollo de habilidades prácticas para utilizar y gestionar información en entornos digitales. A diferencia de la alfabetización informacional que abarca el acceso a la información en cualquier formato, la alfabetización digital implica que las personas adquieran competencias aplicables a lo largo de su vida, tanto en el ámbito educativo, como en el sociolaboral y económico, permitiéndoles resolver problemas, analizar información electrónica y desenvolverse en entornos tecnológicos de manera segura y eficiente (García-Ávila, 2017, p. 71).

API

API significa "interfaz de programación de aplicaciones" y son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos (AWS 2202).

Arquitectura

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución. (Hilliard, R. 2000)

Aseguramiento de la Calidad del Software (SQA)

El aseguramiento de la calidad del software (SQA) comprende un conjunto de actividades diseñadas para garantizar que un producto de software cumpla con los estándares de calidad establecidos. Este proceso se lleva a cabo de manera independiente al equipo de desarrollo y emplea métricas para optimizar el proceso de producción de software, lo que a su vez

contribuye a mejorar la calidad del producto final (Rodríguez Batista & Toiran González, 2007).

Backend

El backend hace referencia al conjunto de tecnologías que operan en el servidor, encargándose de gestionar bases de datos, autenticar usuarios y procesar la lógica del negocio. A diferencia del frontend, el backend permite el uso de diversos lenguajes de programación como PHP, Python, .NET y Java, así como bases de datos como SQL, MongoDB y MySQL (Ferry Mestres, s.f.).

Ciclo de vida de desarrollo de software

El Software Development Life Cycle (SDLC) es una metodología estructurada que guía el desarrollo de software a lo largo de su ciclo de vida, desde la planificación hasta su retiro. Este proceso incluye diversas etapas clave como la recopilación de requisitos, diseño, pruebas, implementación y mantenimiento. En el ámbito de la investigación clínica, el SDLC es crucial para garantizar la calidad y la integridad de los datos, asegurando que los sistemas cumplan con normativas técnicas y regulatorias (Wilson et al., 2024).

CRUD

CRUD hace referencia a cuatro operaciones básicas crear, leer, actualizar y eliminar que permiten administrar la información dentro de distintos sistemas o desarrollos de software (Baumann, 2022).

ERS

Los requisitos se entienden como las condiciones o necesidades que tiene un usuario para poder solucionar un problema o cumplir un objetivo (Institute of Electrical and Electronics Engineers, 1990).

Framework

Un framework es una estructura base de soporte que facilita el desarrollo de software, proporcionando un conjunto de funciones y código genérico para tareas comunes como la creación de objetos y conexión a bases de datos. Su uso permite a los desarrolladores enfocarse

en los requerimientos específicos de cada aplicación, reduciendo tiempos de implementación y garantizando un código modular, reutilizable y escalable (Villalobos, Sánchez & Gutiérrez, 2010).

Frontend

El frontend comprende las tecnologías web que se ejecutan en el navegador del usuario, permitiendo la interacción con la interfaz gráfica. Entre los lenguajes esenciales de este desarrollo se encuentran HTML, CSS y JavaScript, los cuales definen la estructura, el diseño y la funcionalidad del sitio web (Ferry Mestres, s.f.).

HTTP / HTTPS

El HTTPS es lo mismo que HTTP solo con una capa más de cifrado que, inicialmente era SSL, es un protocolo para transmitir la información segura por la web (Galhego, 2022).

Inmuebles

Es una propiedad (ya sea un terreno, una casa u otro tipo de inmueble, ubicado en el campo o en la ciudad) que se entrega como parte de una dote es decir, como aportación económica al matrimonio ya sea de manera obligatoria o voluntaria. Este bien puede tener un valor previamente acordado (estimado) o no tener un valor fijo establecido (Cabanellas, G., & Cabanellas, G. 1979).

Laravel

Laravel es un framework de código abierto ampliamente utilizado en el desarrollo web, que destaca por facilitar el proceso de programación mediante diversas características como la modularidad, el enrutamiento HTTP ágil, el control de accesos mediante Middleware, autenticación nativa y un sistema de encriptación basado en OpenSSL y TLS. Su adaptabilidad con Lumen permite además la creación de microservicios y APIs, convirtiéndolo en una herramienta flexible y segura para el desarrollo de aplicaciones web (Avilés, Avila-Pesantez & Avila, 2020).

PHP

PHP es un lenguaje de programación web que se caracteriza por poder combinarse con HTML, siendo ampliamente utilizado en el desarrollo de sitios y aplicaciones web dinámicas (Fernández & Rodríguez, 2021).

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional que supera las limitaciones de los sistemas relacionales tradicionales, incorporando conceptos como clases, herencia, tipos y funciones, lo que permite a los usuarios extender el sistema de forma flexible, añadiendo además características como restricciones, disparadores, reglas e integridad transaccional (Lockhart, 1996).

Scrum

Scrum es una metodología ágil de desarrollo de software basada en ciclos cortos de trabajo denominados Sprints, que promueve la auto-organización de equipos multidisciplinares para gestionar proyectos en entornos de incertidumbre. Surgida en 1993 con Jeff Sutherland y formalizada en 1996 junto a Ken Schwaber, esta metodología permite desarrollar productos de forma flexible e iterativa, priorizando la colaboración, la transmisión del conocimiento y la entrega progresiva de resultados (Gallego, 2012).

UI/UX

El diseño UI/UX es clave en los productos digitales, ya que influye en la interacción del usuario; una buena interfaz facilita la navegación y una adecuada experiencia mejora la satisfacción y el compromiso (Hamidli, 2023).

XP (Extreme Programming)

Extreme Programming XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo (Letelier & Penadés, 2006).

Supabase

Supabase es una plataforma de desarrollo lanzada en 2020 que integra múltiples servicios de backend en un solo entorno, incluyendo autenticación de usuarios, gestión de bases de datos, almacenamiento de archivos, generación automática de APIs y funciones de borde. Su biblioteca cliente es compatible con lenguajes como JavaScript, Python y Dart (Ayezabu, 2022).

2.4 Legal

Este marco legal se basa en la siguiente reglamentación:

2.4.1 Ley de Inquilinato

La Ley de Inquilinato establece el marco regulatorio para las relaciones contractuales de arrendamiento en zonas urbanas del Ecuador.

El artículo 1 delimita el alcance territorial de la norma a los perímetros urbanos definidos por ordenanza municipal (Congreso Nacional del Ecuador, 2000, art. 1), lo que aplica directamente al contexto de la parroquia Guanujo.

El artículo 3 define las condiciones mínimas que debe cumplir todo inmueble destinado al arrendamiento, entre ellas servicios higiénicos, aireación, agua potable y ausencia de riesgo estructural. Estas condiciones son registradas y monitoreadas por el sistema a través del módulo de incidencias (Congreso Nacional del Ecuador, 2000, art. 3).

El artículo 4 obliga al arrendador a ejecutar las reparaciones necesarias para mantener dichas condiciones, responsabilidad que el sistema facilita mediante la gestión de tickets de mantenimiento (Congreso Nacional del Ecuador, 2000, art. 4).

Por su parte, el artículo 7 establece que el arrendatario responde por los daños que cause en el inmueble, lo que justifica la inclusión del registro fotográfico de evidencias dentro del módulo de tickets (Congreso Nacional del Ecuador, 2000, art. 7).

Respecto a los contratos, el artículo 27 reconoce validez tanto a los acuerdos verbales como a los escritos, y el sistema formaliza digitalmente ambos tipos (Congreso Nacional del Ecuador, 2000, art. 27).

El artículo 29 exige que los contratos cuyo valor supere un salario básico unificado se celebren por escrito y se registren ante notario. Esta obligación es apoyada por el módulo de contratos digitales, el cual genera documentos descargables (Congreso Nacional del Ecuador, 2000, art. 29).

Finalmente, el artículo 30 enumera las causales de terminación del contrato, entre ellas la falta de pago de dos mensualidades consecutivas. Esta causal es detectada automáticamente por el sistema mediante el panel de seguimiento de cobros (Congreso Nacional del Ecuador, 2000, art. 30).

2.4.2 Ley Orgánica de Protección de Datos Personales

Esta ley regula el tratamiento de datos personales en Ecuador y es directamente aplicable al sistema, puesto que este gestiona información sensible de dos grupos vulnerables a riesgos de privacidad: estudiantes universitarios y propietarios de inmuebles.

El artículo 42 exige que el responsable del tratamiento realice evaluaciones de impacto cuando exista riesgo alto para los derechos del titular (Asamblea Nacional del Ecuador, 2021, art. 42). Esta evaluación se materializó en la arquitectura de roles diferenciados del sistema, donde cada usuario accede únicamente a la información que le corresponde según su perfil.

El artículo 43 impone una obligación concreta: notificar cualquier vulneración de seguridad a la Autoridad de Protección de Datos en un plazo máximo de 5 días (Asamblea Nacional del Ecuador, 2021, art. 43). Detrás de ese plazo hay una exigencia técnica implícita sin capacidad de detección oportuna, la notificación no es viable. El sistema responde a esta exigencia desde cuatro frentes. Primero, los logs de actividad registran de manera continua todas las operaciones sobre los datos, generando una trazabilidad que permite identificar el momento exacto y el alcance de cualquier incidente. Segundo, el cifrado de información sensible en la base de datos garantiza que, incluso ante una brecha, los datos comprometidos no sean legibles. Tercero, el sistema de notificaciones automáticas informa de inmediato al administrador responsable cuando se detecta un evento de seguridad crítico, acortando el tiempo de respuesta. Cuarto, el control de accesos por roles restringe la exposición de datos a los usuarios estrictamente autorizados, reduciendo la superficie de riesgo desde el diseño mismo del sistema.

El artículo 53 promueve la elaboración de códigos de conducta por parte de organizaciones para garantizar el cumplimiento normativo (Asamblea Nacional del Ecuador, 2021, art. 53). Este principio se traduce en las políticas de privacidad implementadas dentro de la plataforma.

2.4.3 Código Orgánico de Economía Social de los Conocimientos (INGENIOS)

Este código respalda jurídicamente la decisión tecnológica del proyecto.

El artículo 142 define el software de código abierto y garantiza al usuario libertades esenciales como ejecutar, estudiar, modificar y redistribuir el código fuente (Asamblea Nacional del Ecuador, 2016, art. 142). Esto valida el uso de PHP, Laravel y PostgreSQL como herramientas libres en este proyecto.

El artículo 144 establece que las instituciones de educación superior deben priorizar el uso de tecnologías digitales libres en su funcionamiento administrativo (Asamblea Nacional del Ecuador, 2016, art. 144). Esta disposición alinea el desarrollo del sistema con las políticas tecnológicas de la Universidad Estatal de Bolívar.

Para el desarrollo de este sistema web no existe impedimento legal alguno. Este proyecto carece de restricciones normativas. El sustento jurídico se articula mediante cuatro ejes fundamentales que garantizan la viabilidad del software.

El Código Orgánico de Economía Social de los Conocimientos faculta expresamente el uso de tecnologías libres. El artículo ciento cuarenta y dos garantiza libertades esenciales para ejecutar, estudiar y adaptar el código fuente según necesidades administrativas específicas. Esta autonomía técnica asegura la innovación al igual que el artículo ciento cuarenta y cuatro que promueve el uso de software de código abierto en instituciones educativas, dando la validación a que se implementen de herramientas de software para optimizar procesos internos sin vulnerar derechos de propiedad intelectual o autoría.

La Ley de Inquilinato admite diversas formas contractuales como el artículo veintisiete que otorga validez legal tanto a los contratos escritos como a los verbales, el sistema formaliza estos tipos de acuerdos de manera que facilita el cumplimiento de las condiciones de habitabilidad exigidas en el artículo tercero, permitiendo que el arrendador realice las reparaciones necesarias de forma oportuna. En el artículo treinta se establecen las causas de terminación del contrato, y unos de los principales motivos es la falta de pagos, las cuales el

aplicativo monitorea mediante el control automático de morosidad y pensiones locativas pendientes.

La seguridad de la información es prioritaria por lo que se aplican rigurosamente los protocolos de la Ley Orgánica de Protección de Datos Personales para salvaguardar la privacidad de estudiantes y propietarios. El artículo cuarenta y dos exige evaluaciones de impacto técnico para mitigar riesgos en el tratamiento de datos sensibles por lo que la integridad de los datos está resguardada.

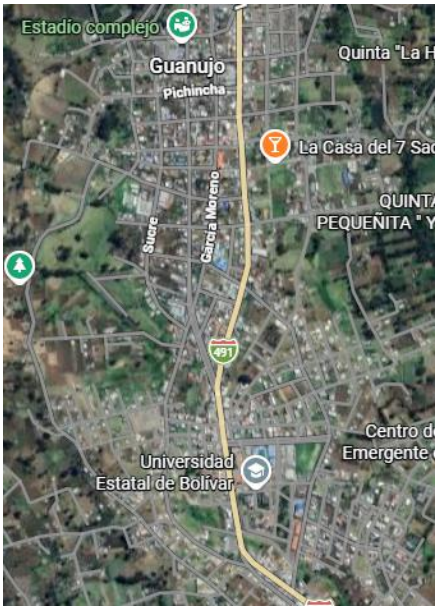
2.5 Georreferencial

El tema de trabajo de titulación e integración curricular se lo aplicó en la Parroquia Guanujo de la ciudad de Guaranda en la provincia de Bolívar, ubicada geográficamente en:

Longitud: 1°34'04.9"S

Latitud: 79°00'29.5"W

Figura 1: Imagen satelital de la parroquia de Guanujo.



Fuente: Google. (2026). *Imagen satelital de la parroquia de Guanujo*. Google Maps. https://maps.google.com (accedido el 12 de abril de 2026).

CAPÍTULO III

METODOLOGÍA

3.1 Metodología de Desarrollo de Software

El presente proyecto de desarrollo del Sistema Web Integral para la Gestión Administrativa, Financiera y de Incidencias en Inmuebles de Alquiler para Estudiantes de la Universidad Estatal de Bolívar adoptó la metodología ágil Extreme Programming (XP) como marco de trabajo para su planificación, desarrollo e implementación.

Aunque XP nació para equipos grandes, sus principios son ideales para este proyecto. Permiten mejorar la eficiencia y la calidad del código mientras el sistema crece. La elección de XP responde a puntos clave que aseguran el éxito de la plataforma:

- Ingeniería gradual y controlada: El sistema se construye paso a paso. Realizar cambios incrementales ayuda a minimizar riesgos, especialmente en los módulos financieros y de pagos.
- Adaptación a cambios: El mercado de alquileres para estudiantes es dinámico. XP permite ajustar los requisitos sobre la marcha sin frenar el progreso.
- Entrega continua de valor: Se priorizan las funciones más urgentes. Esto permite tener versiones funcionales de GestInWEB en ciclos muy cortos.
- Flexibilidad total: Cada iteración es una oportunidad para ajustar detalles según las necesidades reales que se detecten durante el desarrollo.

3.1.1 Adaptación de XP para un desarrollador individual

Las prácticas de XP se han ajustado para optimizar el trabajo en solitario dentro del sistema de GestInWEB. Las principales adaptaciones son:

Desarrollo iterativo e incremental: El trabajo se divide en iteraciones de una a dos semanas. Cada etapa tiene metas claras, como terminar el módulo de incidencias o la gestión de contratos. Al cerrar cada ciclo, se validan los avances para asegurar que todo funciona según lo previsto.

- Diseño simple y evolutivo: Se aplica un enfoque de diseño mínimo. El objetivo es que GestInWEB sea funcional desde el inicio sin añadir complicaciones técnicas innecesarias.

El sistema evolucionará de forma orgánica según las necesidades de los arrendatarios y dueños de inmuebles.

- Refactorización y rigor técnico: El código se revisa y limpia de forma permanente. Esto aumenta la legibilidad y facilita el mantenimiento a futuro, y también se priorizan cambios pequeños para evitar que aparezcan fallos inesperados en la base de datos o la interfaz.
- Integración continúa adaptada: Se utiliza Git para gestionar cada avance. Cada cambio se valida de inmediato para detectar errores de forma temprana de manera de que se garantiza que la plataforma sea estable en todo momento.
- Pruebas automáticas: Se implementan pruebas para las funciones críticas del sistema. Esto funciona como un respaldo de seguridad que permite modificar el código con total confianza.

Esta adaptación de XP permite aprovechar al máximo la agilidad del proceso. Asegura que GestInWEB evolucione de manera segura, manteniendo un desarrollo eficiente y una estabilidad constante en cada entrega.

3.1.2 Faces de la Metodología Extreme Programming XP

La metodología XP organiza el desarrollo mediante ciclos ágiles y rigurosos. Según Kent Beck el creador de esta metodología, el proceso se divide en cuatro etapas fundamentales que priorizan la comunicación y la calidad técnica.

Fase I - Planificación del proyecto

La metodología XP comienza con la definición de historias de usuario, que son descripciones cortas y simples que el cliente escribe directamente para estimar el tiempo de desarrollo y no trabajar arduamente en explicar los algoritmos o la estructura de la base de datos o los detalles técnicos del software. Basándose en esa información, el equipo técnico desarrolla un plan de lanzamiento que establece las fechas, prioridades y recursos asignados a cada versión del software.

A partir de ahora, el trabajo se organiza en iteraciones de tres semanas, en las cuales se selecciona el trabajo crítico y se mide la velocidad del proyecto para una entrega más ágil y constante. La programación se realiza en parejas en la misma estación de trabajo, donde un desarrollador escribe el código y el otro analiza el diseño y detecta errores en tiempo real. Este

proceso se sincroniza a través de reuniones diarias cortas donde el equipo plantea desafíos y propone soluciones en conjunto.

Fase II - Diseño

El diseño debe permanecer siempre simple. Evitar complicaciones innecesarias reduce esfuerzos futuros y facilita la implementación inmediata. Los desarrolladores utilizan glosarios de términos. Estandarizar nombres de métodos y clases ayuda a comprender el sistema global. Si surgen riesgos altos, una pareja investiga el problema profundamente. Nunca añadimos funcionalidad extra al programa. Desarrollar herramientas innecesarias desperdicia tiempo valioso del equipo. El equipo refactoriza el código constantemente. Mejorar las estructuras existentes optimiza el funcionamiento sin alterar el comportamiento externo de la aplicación.

Fase III - Codificación

La codificación requiere la presencia constante del cliente. Él detalla funciones específicas y valida los resultados finales del software. Los programadores siguen estándares de codificación estrictos. Programar bajo normas preestablecidas garantiza un código consistente, comprensible y escalable. Esta disciplina facilita que cualquier integrante del equipo pueda modificar el repositorio compartido sin dificultades técnicas.

Fase IV - Pruebas

Las pruebas automatizadas aseguran la calidad técnica superior. Los desarrolladores crean test específicos antes de codificar la lógica del negocio. Estas herramientas evalúan las clases del sistema omitiendo únicamente los métodos triviales. Los test no deben tener dependencias externas del código evaluado. El equipo realiza test de aceptación finales. Estas pruebas verifican que las funcionalidades generales cumplan los requisitos especificados por el cliente. El programa debe superar cada evaluación para considerarse terminado.

Figura 1. Fases de la Metodología XP



Fuente: *Bustamante y Rodríguez, (2014)*

3.2 Técnicas e Instrumentos de Recopilación de Datos

Para el desarrollo del sistema web integral y el levantamiento de requerimientos, se implementó diversas técnicas de recopilación de datos con el fin de obtener información precisa sobre la problemática de la gestión de alquileres en la parroquia Guanujo. El uso de un enfoque mixto (cualitativo y cuantitativo) permitió conocer tanto las necesidades logísticas de los propietarios como las expectativas de los estudiantes de la Universidad Estatal de Bolívar.

3.2.1 Entrevista

La entrevista cualitativa permite profundizar en procesos complejos. Estas dinámicas suelen escapar a las opciones cerradas de las encuestas tradicionales. Su efectividad nace del diálogo directo con los clientes del producto. Se realizaron entrevistas a propietarios con numerosos inquilinos y también a los estudiantes que arriendan. Estos testimonios aclararon el panorama socioeconómico y legal de los arrendamientos. Se utilizó un guion de entrevista semiestructurado, preguntas flexibles que abarcaron temas emergentes sobre la seguridad de los pagos y la formalidad de los contratos. ANEXO 3

3.2.2 Encuesta

La encuesta fue la técnica cuantitativa principal de este estudio. Permitió recolectar datos masivos y estandarizados sobre el comportamiento, preferencias y dificultades de los actores involucrados. Se aplicó a dos grupos focales: **arrendatarios (estudiantes)** para medir su experiencia de usuario y **arrendadores (propietarios)** para definir la lógica de negocio administrativa. ANEXO 3

CAPITULO IV

INGENIERÍA DEL PROYECTO

El desarrollo técnico del sistema web, estructurado según las fases de la metodología Extreme Programming (XP), la cual permite construir el sistema de forma iterativa, validando cada módulo con los usuarios reales antes de avanzar al siguiente. Esto es clave. En un contexto donde la informalidad predomina y los requerimientos pueden ajustarse sobre la marcha, una metodología rígida habría limitado la capacidad de respuesta frente a los cambios constantes del desarrollo. A lo largo de este capítulo se detallan las fases de planificación, diseño, codificación y pruebas, cada una aplicada al desarrollo del Sistema Integral para la Gestión Administrativa, Financiera y de Incidencias en inmuebles de alquiler para estudiantes de la Universidad Estatal de Bolívar.

4.1 Fase I: Planificación

La fase de planificación parte del análisis de las encuestas y entrevistas aplicadas a arrendatarios y arrendadores de la parroquia Guanujo, cuyos resultados permitieron identificar las necesidades reales de cada actor del sistema. Con esa base, se redactaron las historias de usuario siguiendo los lineamientos de la metodología Extreme Programming (XP), donde cada historia describe una funcionalidad concreta desde la perspectiva del usuario. A partir de ellas se construyó el plan de entregas, definiendo prioridades, tiempos estimados e iteraciones de desarrollo.

4.1.1 Universo

La población de estudio para el presente proyecto está conformada por los actores principales del sector de alquileres en la parroquia Guanujo, específicamente estudiantes de la Universidad Estatal de Bolívar y propietarios de inmuebles.

Tabla 3. Número de participantes de la encuesta

Descripción	**Cantidad**
Arrendatarios (Estudiantes UEB)	50
Arrendadores (Propietarios de inmuebles)	25
Total	**75**

Nota. Elaborado por: Andrade P. Kevin.

La población seleccionada es de tipo no probabilística por conveniencia, centrada en 50 estudiantes que residen en la zona y 25 arrendadores, quienes proporcionaron la información necesaria sobre los procesos administrativos, financieros y de incidencias para el diseño y validación del sistema web integral.

4.1.2 Procesamiento de datos

Para el procesamiento de la información se aplicó un enfoque cuantitativo mediante técnicas de estadística descriptiva ,tomando los resultados obtenidos a través de encuestas digitales. El flujo de procesamiento se detalla a continuación:

- **Tabulación:** Los datos fueron organizados de manera tabular, permitiendo la clasificación de frecuencias y el cálculo de porcentajes para cada variable.
- **Representación Gráfica:** Se generaron gráficos estadísticos en formato circular 3D para facilitar la interpretación visual de los resultados.
- **Análisis:** Cada gráfico se le realizó un análisis descriptivo para identificar los puntos críticos y requerimientos técnicos que fundamentan el desarrollo del sistema web.

4.1.2.1 Análisis de Datos

Encuesta a Arrendatarios

Sección 1: Perfil y Gestión Administrativa

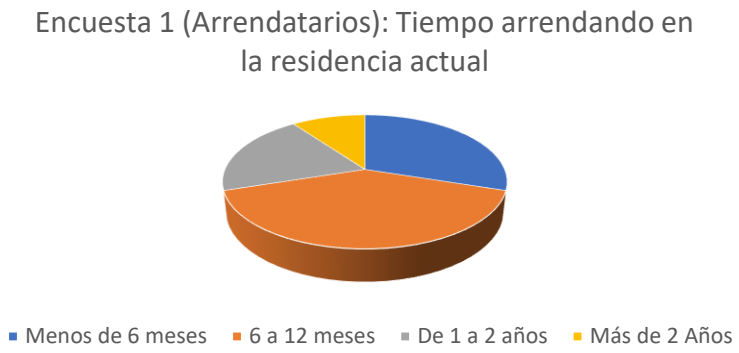
¿Cuánto tiempo lleva arrendando en tu residencia actual?

Tabla 4. Tabulación de datos de la pregunta 1 a arrendatarios

Opción	Frecuencia	Porcentaje
Menos de 6 meses	15	30%
6 a 12 meses	20	40%
1 a 2 años	10	20%
Más de 2 años	5	10%
Total	**50**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 2. ***Diagrama de la tabulación de datos de la pregunta 1 a arrendatarios***



Nota. Elaborado por: Andrade P. Kevin (2026).

Los datos recolectados revelan una movilidad residencial elevada entre los estudiantes universitarios. El cuarenta por ciento de los participantes habita su residencia actual entre seis y doce meses. Esta cifra coincide usualmente con la duración de un ciclo académico ordinario. Un treinta por ciento adicional de la muestra registra una estancia inferior a un semestre. Estas tendencias demuestran que la mayoría de los arrendatarios son usuarios temporales o recién llegados. Pocos estudiantes mantienen contratos extensos. Solo el diez por ciento de los encuestados supera los dos años de permanencia continua en el mismo inmueble. Puede haber muchos factores por lo que la permanencia de los inquilinos en los inmuebles es corta, puede ser problemas con otros inquilinos, incumplimiento de los mantenimientos o reparaciones del inmueble por parte del arrendador, morosidad en los pagos, etc.

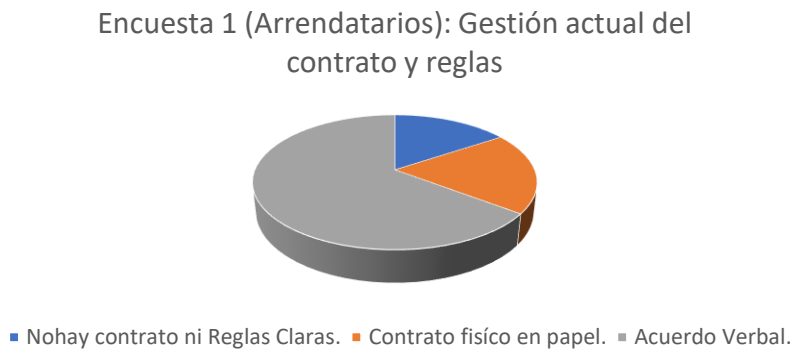
Seleccione: Actualmente ¿cómo gestiona el contrato y las reglas de tu alojamiento?

Tabla 5. ***Tabulación de datos de la pregunta 2 a arrendatarios***

Opción	**Frecuencia**	**Porcentaje**
Contrato físico firmado en papel	10	20%
Acuerdo verbal únicamente	32	64%
No cuento con contrato ni reglas claras	8	16%
Total	**50**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 3. Diagrama de la tabulación de datos de la pregunta 2 a arrendatarios



Nota. Elaborado por: Andrade P. Kevin (2026).

El 64% de los participantes utiliza acuerdos verbales únicamente para gestionar su estancia, esta cifra equivale a 32 estudiantes. La informalidad prevalece en la parroquia Guanujo. Estos pactos carecen de respaldo físico y la ausencia de documentos genera inseguridad jurídica para ambas partes, a su vez solo 10 personas firman contratos físicos en papel lo cual representa una minoría del 20%. El uso de registros impresos resulta insuficiente para las demandas actuales, y la gestión análoga dificulta la consulta inmediata de las normas de convivencia. Por eso, el sistema web debe digitalizar estos procesos para garantizar la transparencia.

Un dato preocupante es que solo 8 estudiantes admiten no poseer contrato ni reglas claras siendo el 16% que vive en una incertidumbre administrativa absoluta. La desorganización impera en estos casos. No existen límites definidos para la habitabilidad.

Seleccione: ¿Qué información le gustaría tener siempre disponible de forma digital?

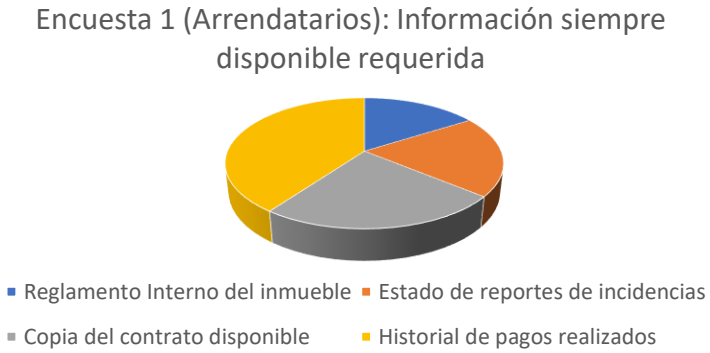
Tabla 6. Tabulación de datos de la pregunta 3 a arrendatarios

Opción	Frecuencia	Porcentaje
Copia digital del contrato	12	24%
Historial de pagos realizados	25	50%
Reglamento interno del inmueble	5	10%
Estado de reportes de mantenimiento	8	16%

Total	**50**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 4. ***Diagrama de la tabulación de datos de la pregunta 3 a arrendatarios***



Nota. Elaborado por: Andrade P. Kevin (2026).

El 50% de los arrendatarios prioriza el acceso inmediato al historial de sus pagos realizados. Esta demanda mayoritaria refleja la necesidad crítica de sustituir los registros manuales por evidencias digitales trazables que eviten conflictos con los propietarios. Asimismo, el 24% de los estudiantes considera fundamental disponer de una copia digital del contrato para garantizar su seguridad jurídica en cualquier momento sin depender de documentos físicos. El interés restante se distribuye entre el seguimiento de mantenimientos y la consulta de reglamentos internos, opciones que acumulan un dieciséis y diez por ciento de la frecuencia total respectivamente. Aunque estos valores son menores frente a la gestión de pagos, su presencia confirma que el sistema debe ofrecer una solución que integre estas cuatro opciones.

Sección 2: Gestión Financiera

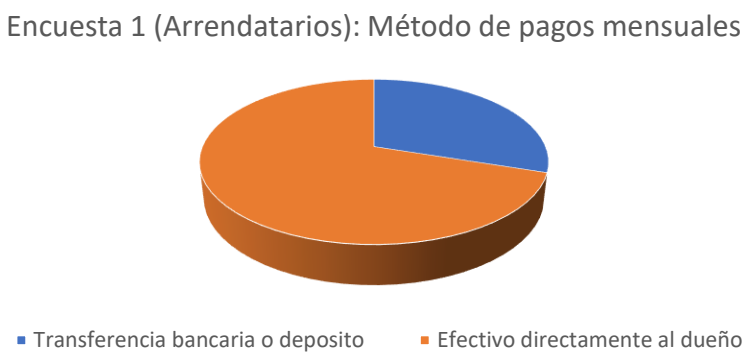
Seleccione: ¿Cómo realiza comúnmente el pago de su mensualidad?

Tabla 7. ***Tabulación de datos de la pregunta 4 a arrendatarios***

Opción	**Frecuencia**	**Porcentaje**
Efectivo directamente al dueño	35	70%
Transferencia bancaria o depósito	15	30%
Total	**50**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 5. ***Diagrama de la tabulación de datos de la pregunta 4 a arrendatarios***



Nota. Elaborado por: Andrade P. Kevin (2026).

El análisis de los resultados demuestra que el 70% de los arrendatarios realiza sus pagos en efectivo directamente al propietario, lo cual confirma la persistencia de procesos informales en la gestión de alquileres de la parroquia Guanujo. Esta cifra representa a 35 de los 50 estudiantes encuestados y evidencia un riesgo administrativo alto por la falta de respaldos financieros automáticos. En contraste, apenas el 30% utiliza transferencias bancarias o depósitos para cumplir sus obligaciones, lo que refleja una brecha digital significativa que el sistema web propuesto debe cerrar mediante la automatización de cobros.

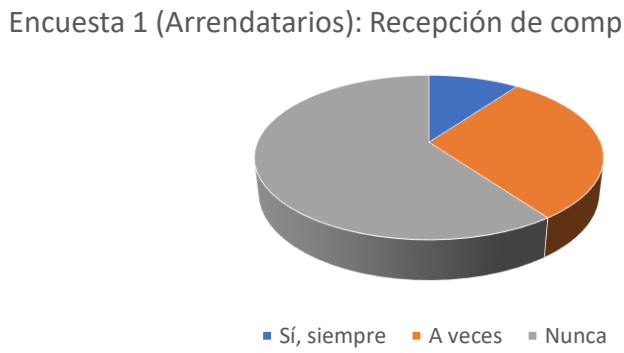
Seleccione: ¿Recibe algún comprobante de pago formal (físico o digital) cada mes?

Tabla 8. ***Tabulación de datos de la pregunta 5 a arrendatarios***

Opción	Frecuencia	Porcentaje
Sí, siempre	5	10%
A veces	15	30%
Nunca	30	60%
Total	**50**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 6. ***Diagrama de la tabulación de datos de la pregunta 5 a arrendatarios***



Nota. Elaborado por: Andrade P. Kevin (2026).

Los resultados estadísticos revelan una carencia crítica de formalidad administrativa, puesto que el 60% de los arrendatarios encuestados nunca recibe un comprobante de pago por su mensualidad. Esta alarmante cifra representa a 30 estudiantes desprotegidos. Al considerar que el 30% adicional obtiene respaldos solo de manera ocasional, el 90% de la población total en Guanujo habita bajo una incertidumbre financiera preocupante. Solo 5 personas disponen siempre de un justificativo legal, esta carencia de respaldo legal justifica el módulo de generación automática de recibos PDF que ofrece el sistema.

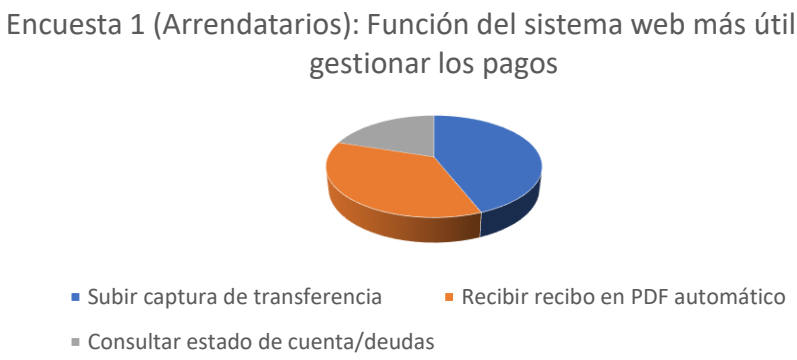
Seleccione: ¿Qué función del sistema web le resultaría más útil para gestionar sus pagos?

Tabla 9. Tabulación de datos de la pregunta 6 a arrendatarios

Opción	**Frecuencia**	**Porcentaje**
Subir captura de transferencia	22	44%
Recibir recibo en PDF automático	18	36%
Consultar estado de cuenta/deudas	10	20%
Total	**50**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 7. Diagrama de la tabulación de datos de la pregunta 6 a arrendatarios



Nota. Elaborado por: Andrade P. Kevin (2026).

El análisis estadístico revela que el 44% de los arrendatarios prefiere la opción de subir capturas de transferencia para validar sus pagos efectuados, puesto que 22 estudiantes identifican esta función como la herramienta más crítica para asegurar la transparencia en sus transacciones financieras. Esta tendencia hacia la verificación digital inmediata se complementa con el 36% de la población que prefiere recibir recibos automáticos en formato PDF para obtener respaldos legales instantáneos sin depender de la gestión manual del propietario, mientras que el 20% restante considera fundamental consultar el estado de sus pagos para visualizar las deudas pendientes en tiempo real.

Seleccione: ¿Le gustaría recibir un recordatorio automático antes de la fecha límite de pago?

Tabla 10. Tabulación de datos de la pregunta 7 a arrendatarios

Opción	**Frecuencia**	**Porcentaje**
Sí	42	84%
No	3	6%
Indiferente	5	10%
Total	**50**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 8. Diagrama de la tabulación de datos de la pregunta 7 a arrendatarios



Nota. Elaborado por: Andrade P. Kevin (2026).

El análisis estadístico evidencia una necesidad latente de puntualidad tecnológica, pues el 84% de los arrendatarios encuestados aprueba la recepción de recordatorios automáticos previos al vencimiento de sus mensualidades. Esta cifra de respuestas involucra a 42 de los 50 participantes que justifica la creación de un sistema de alertas que organice el flujo financiero estudiantil, eliminando el olvido de los pagos que derivan en conflictos administrativos. Los datos muestran que solo 3 personas rechazan esta herramienta y 5 mantienen una postura indiferente, proporciones mínimas que no restan valor a la alta demanda por automatizar las notificaciones de cobro.

Sección 3: Gestión de Incidencias

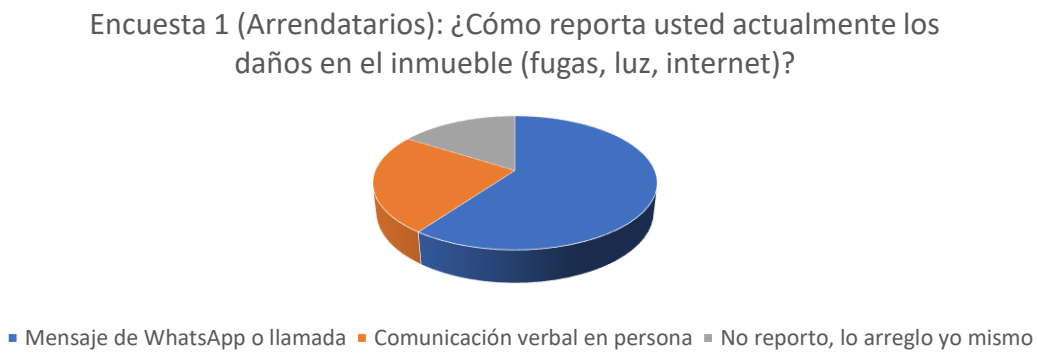
Seleccione: ¿Cómo reporta usted actualmente los daños en el inmueble (fugas, luz, internet)?

Tabla 11. Tabulación de datos de la pregunta 8 a arrendatarios

Opción	**Frecuencia**	**Porcentaje**
Mensaje de WhatsApp o llamada	30	60%
Comunicación verbal en persona	12	24%
No reporto, lo arreglo yo mismo	8	16%
Total	**50**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 9. Diagrama de la tabulación de datos de la pregunta 8 a arrendatarios



Nota. Elaborado por: Andrade P. Kevin (2026).

El 60% de los arrendatarios utiliza medios informales como WhatsApp o llamadas telefónicas para reportar averías, lo cual genera una falta de seguimiento técnico sobre el estado real de los inmuebles en la parroquia Guanujo. Esta cifra representa a 30 estudiantes. Puesto que el 24% adicional depende de la comunicación verbal directa, la ausencia de registros formales dificulta la planificación de reparaciones urgentes y compromete la habitabilidad de las viviendas. Resulta preocupante que el 16% de los jóvenes prefiera solucionar los desperfectos por cuenta propia, conducta que oculta el deterioro progresivo de la infraestructura al propietario.

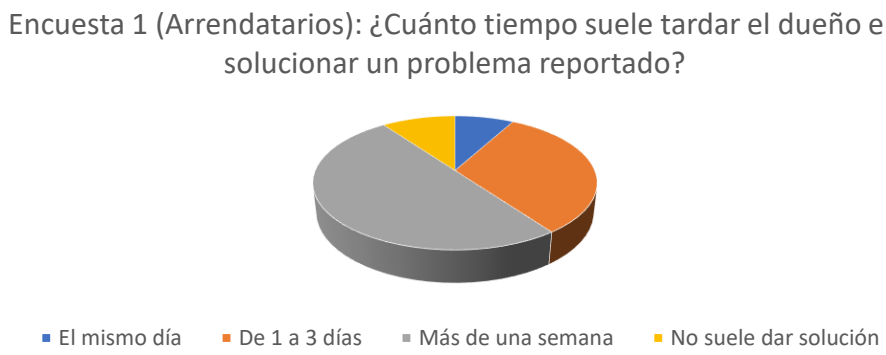
Seleccione: ¿Cuánto tiempo suele tardar el dueño en solucionar un problema reportado?

Tabla 12. Tabulación de datos de la pregunta 9 a arrendatarios

Opción	**Frecuencia**	**Porcentaje**
El mismo día	4	8%
De 1 a 3 días	16	32%
Más de una semana	25	50%
No suele dar solución	5	10%
Total	**50**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 10. Diagrama de la tabulación de datos de la pregunta 9 a arrendatarios



Nota. Elaborado por: Andrade P. Kevin (2026).

La deficiencia en la gestión de mantenimientos es preocupante, puesto que el 50% de los arrendatarios encuestados espera más de una semana para que el propietario solucione los desperfectos reportados en su vivienda. Este largo retraso afecta a 25 estudiantes. El hecho de que el 8% de todos los casos se aborden el mismo día muestra la falta de canales operativos efectivos para resolver inmediatamente los problemas urgentes de habitabilidad. La mayoría de los inquilinos viven con daños técnicos durante períodos prolongados y la situación empeora cuando el 10% de los inquilinos informa que el propietario no responde en absoluto a sus quejas. Solo 16 personas sienten una gestión moderada al recibir reparaciones incluso dentro de uno a tres días.

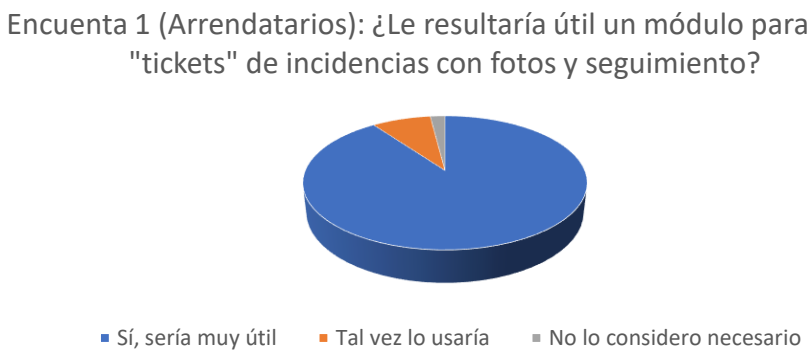
Seleccione: ¿Le resultaría útil un módulo para crear "tickets" de incidencias con fotos y seguimiento?

Tabla 13. Tabulación de datos de la pregunta 10 a arrendatarios

Opción	**Frecuencia**	**Porcentaje**
Sí, sería muy útil	45	90%
Tal vez lo usaría	4	8%
No lo considero necesario	1	2%
Total	**50**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 11. Diagrama de la tabulación de datos de la pregunta 10 a arrendatarios



Nota. Elaborado por: Andrade P. Kevin (2026).

Hay una aceptación abrumadora hacia la modernización del mantenimiento, puesto que el 90% de los encuestados califica como muy útil la implementación de un módulo de tickets con soporte multimedia. Esta preferencia mayoritaria involucra a 45 estudiantes decididos a documentar visualmente cualquier desperfecto técnico para obtener respuestas inmediatas y verificables, lo cual contrasta con el apenas 8% que muestra dudas sobre su uso o el escaso 2% que lo considera innecesario. Al integrar fotos y seguimiento en tiempo real, el sistema web resolverá la falta de registros formales que actualmente entorpece el seguimiento para la resolución de estos incidentes.

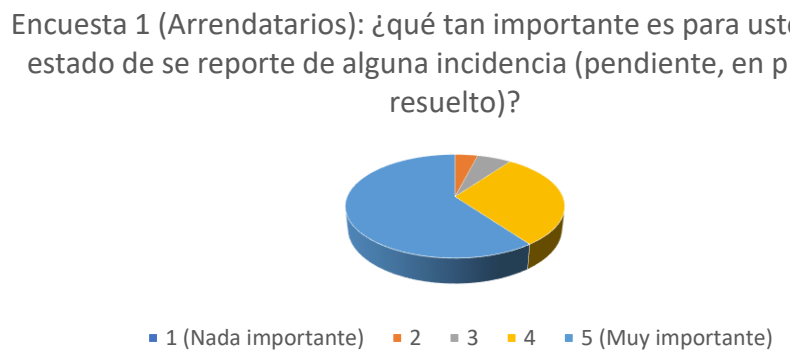
Seleccione: En una escala del 1 al 5, ¿qué tan importante es para usted ver el estado de su reporte (pendiente, en proceso, resuelto)?

Tabla 14. Tabulación de datos de la pregunta 11 a arrendatarios

Opción	**Frecuencia**	**Porcentaje**
1 (Nada importante)	0	0%
2	2	4%
3	3	6%
4	15	30%
5 (Muy importante)	30	60%
Total	**50**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 12. Diagrama de la tabulación de datos de la pregunta 11 a arrendatarios



Nota. Elaborado por: Andrade P. Kevin (2026).

El 90% de los arrendatarios otorga una importancia crítica al seguimiento de sus reportes, puesto que 45 estudiantes seleccionaron las puntuaciones de 4 y 5 para validar la necesidad de visualizar estados como pendiente, en proceso o resuelto a las incidencias reportadas. Esta demanda masiva de transparencia justifica la creación de una interfaz dinámica, mientras que apenas un 4% de los encuestados eligió la opción 2 y ningún participante consideró la herramienta como irrelevante. La integración de indicadores de progreso permitirá que las incidencias cuenten con una trazabilidad técnica verificable, que ayudará al estudiante conocer que su reporte de la incidencia está siendo gestionada.

Encuesta a arrendadores

Sección 1: Gestión Administrativa

Seleccione: ¿Cuántos inquilinos (estudiantes) administra usted en promedio?

Tabla 15. Tabulación de datos de la pregunta 1 a arrendadores

Opción	**Frecuencia**	**Porcentaje**
1 a 5 inquilinos	12	48%
6 a 15 inquilinos	8	32%
Más de 15 inquilinos	5	20%
Total	**25**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 13. Diagrama de la tabulación de datos de la pregunta 1 a arrendadores



Nota. Elaborado por: Andrade P. Kevin (2026).

El análisis estadístico muestra que el 48% de los arrendadores administra entre 1 y 5 inquilinos, mientras que el 32% gestiona grupos de 6 a 15 estudiantes. Estas cifras indican una carga operativa moderada para la mayoría. Sin embargo, el 20% restante supervisa a más de 15 personas lo cual en cuanto aumenta número de inquilinos, mayor es la carga administrativa lo que hace al sistema una herramienta de ahorro de tiempo necesaria.

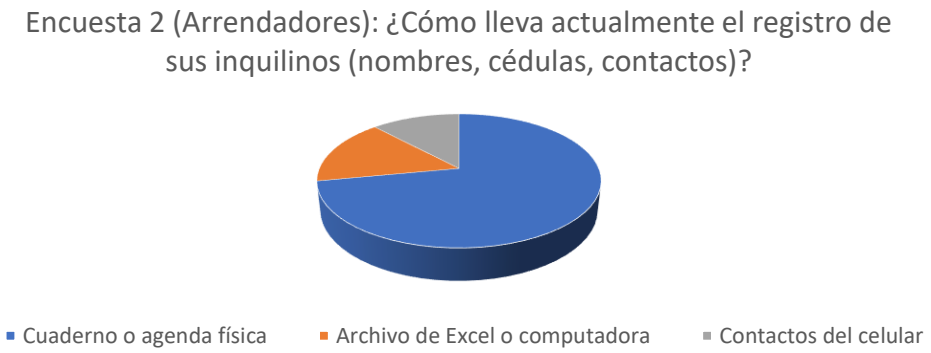
Seleccione: ¿Cómo lleva actualmente el registro de sus inquilinos (nombres, cédulas, contactos)?

Tabla 16. Tabulación de datos de la pregunta 2 a arrendadores

Opción	**Frecuencia**	**Porcentaje**
Cuaderno o agenda física	18	72%
Archivo de Excel o computadora	4	16%
Contactos del celular	3	12%
Total	**25**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 14. Diagrama de la tabulación de datos de la pregunta 2 a arrendadores



Nota. Elaborado por: Andrade P. Kevin (2026).

Los hallazgos demuestran una dependencia significativa de los métodos tradicionales, ya que el 72% de los propietarios todavía utiliza un cuaderno o planificador físico para registrar a sus inquilinos. El enfoque manual crea altos riesgos de pérdida de datos. Solo el 16% utiliza archivos de Excel y solo el 12% usa contactos telefónicos, lo que hace que la migración digital sea una necesidad crítica para mantener la integridad de la información.

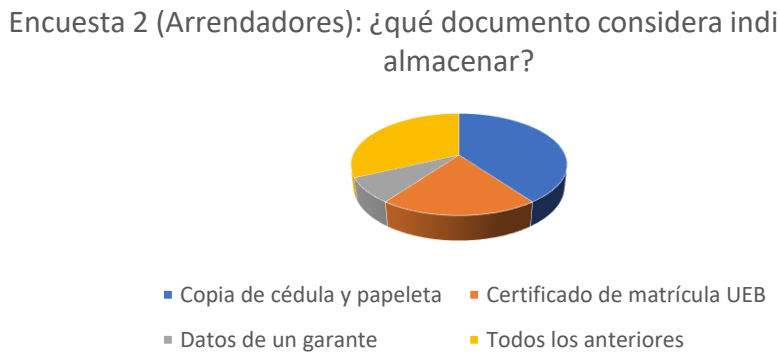
Seleccione: Al registrar un nuevo estudiante, ¿qué documento considera indispensable almacenar?

Tabla 17. Tabulación de datos de la pregunta 3 a arrendadores

Opción	**Frecuencia**	**Porcentaje**
Copia de cédula y papeleta	10	40%
Certificado de matrícula UEB	5	20%
Datos de un garante	2	8%
Todos los anteriores	8	32%
Total	**25**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 15. Diagrama de la tabulación de datos de la pregunta 3 a arrendadores



Nota. Elaborado por: Andrade P. Kevin (2026).

Respecto a la documentación indispensable, el 40% de los encuestados prioriza la copia de cédula y papeleta de votación durante el registro de nuevos estudiantes a su vez otros propietarios muestran requisitos más estrictos, el 32% exige el expediente completo incluyendo certificados de matrícula y datos de garantes. Solamente el 8% que equivale a dos encuestados prefieren solo el número del garante, esta diversidad de criterios exige que el software funcione como un repositorio digital flexible y organizado. Centralizar estos archivos sensibles protegerá legalmente tanto al arrendador como al arrendatario.

Sección 2: Gestión Financiera

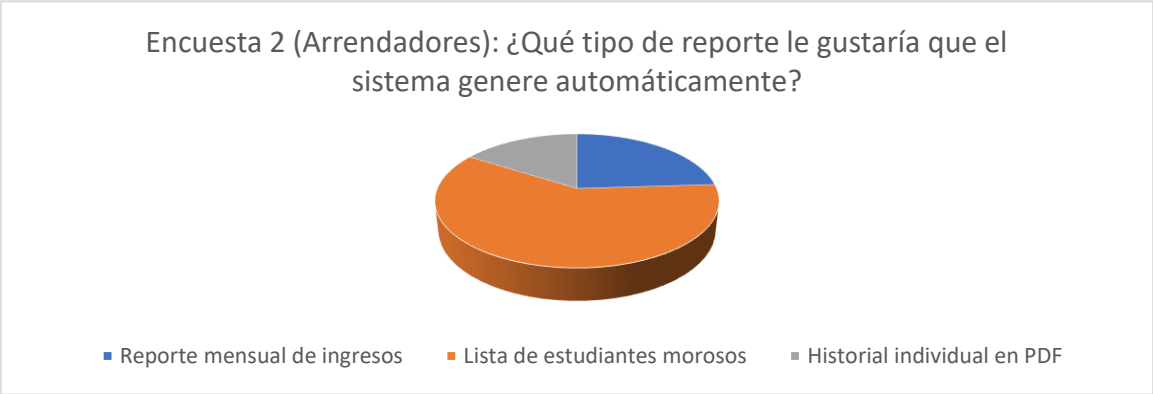
Seleccione: ¿Qué tipo de reporte le gustaría que el sistema genere automáticamente?

Tabla 18. Tabulación de datos de la pregunta 4 a arrendadores

Opción	**Frecuencia**	**Porcentaje**
Reporte mensual de ingresos	6	24%
Lista de estudiantes morosos	15	60%
Historial individual en PDF	4	16%
Total	**25**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 16. Diagrama de la tabulación de datos de la pregunta 4 a arrendadores



Nota. Elaborado por: Andrade P. Kevin (2026).

El 60% de los arrendadores demanda prioritariamente una lista automatizada de estudiantes morosos para ejercer un control financiero más estricto sobre sus activos. Esta función constituye el mayor atractivo del sistema. Además, el 24% requiere reportes mensuales de ingresos y el 16% valora obtener historiales individuales en formato PDF. Los datos confirman que la plataforma debe priorizar la vigilancia de saldos pendientes para asegurar la rentabilidad del negocio inmobiliario en Guanujo.

Sección 3: Gestión de Incidencias y Notificaciones

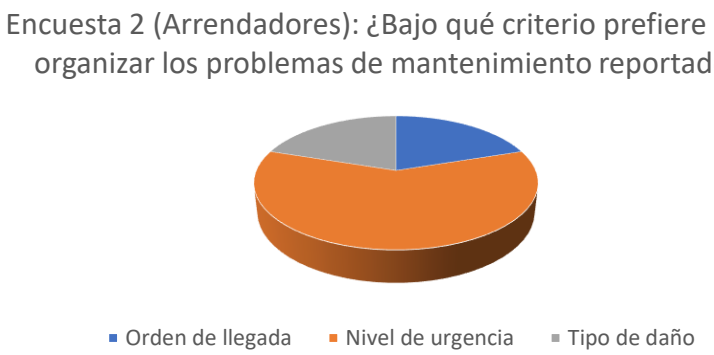
Seleccione: ¿Bajo qué criterio prefiere usted organizar los problemas de mantenimiento reportados?

Tabla 19. Tabulación de datos de la pregunta 5 a arrendadores

Opción	**Frecuencia**	**Porcentaje**
Orden de llegada	5	20%
Nivel de urgencia	15	60%
Tipo de daño	5	20%
Total	**25**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 17. Diagrama de la tabulación de datos de la pregunta 5 a arrendadores



Nota. Elaborado por: Andrade P. Kevin (2026).

Para gestionar el mantenimiento o los incidentes, el 60% de los propietarios prefieren organizar los informes según el nivel de urgencia del daño detectado. Esta priorización inteligente permite abordar primero los problemas críticos que comprometen la estructura de la propiedad. El 20% prefiere manejarlos en el orden en que llegan, y otro 20% prefiere clasificarlos según el tipo de defecto.

Seleccione: ¿Le gustaría que el sistema envíe notificaciones automáticas de recordatorio de pago?

Tabla 20. Tabulación de datos de la pregunta 6 a arrendadores

Opción	**Frecuencia**	**Porcentaje**
Sí, me ahorraría tiempo	22	88%
No, prefiero personalmente	3	12%
Total	**25**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 18. Diagrama de la tabulación de datos de la pregunta 6 a arrendadores



Nota. Elaborado por: Andrade P. Kevin (2026).

La aceptación tecnológica es masiva, pues el 88% de los arrendadores afirma que recibir notificaciones automáticas de recordatorio de pago les ahorraría tiempo valioso. Esta funcionalidad elimina la necesidad de realizar cobros personales que suelen resultar incómodos para ambas partes. Solo 3 personas, que representan el 12%, prefieren mantener el contacto directo con el estudiante.

Seleccione: ¿Estaría dispuesto a actualizar el estado de las reparaciones (ej: "En Proceso" o "Solucionado")?

Tabla 21. Tabulación de datos de la pregunta 7 a arrendadores

Opción	**Frecuencia**	**Porcentaje**
Sí	20	80%
No	5	20%
Total	**25**	**100%**

Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 19. Diagrama de la tabulación de datos de la pregunta 7 a arrendadores



Nota. Elaborado por: Andrade P. Kevin (2026).

Los resultados muestran un dato alentador El 80% de los arrendadores estarían dispuestos a mantener actualizado el estado de las reparaciones que realicen. Este compromiso ayudaría a que la comunicación entre propietarios e inquilinos fluya de forma clara y constante, apoyándose en indicadores sencillos como "En proceso" o "Solucionado", que todos puedan entender de un vistazo. En cambio, un 20% de los participantes se mostró algo más reticente a sumarse al cambio digital. Sin embargo, la mayoría deja clara una idea: contar con un espacio común donde se pueda seguir el camino de que cada arreglo no solo es útil, sino que aporta un beneficio directo a toda la comunidad universitaria.

4.1.2.2 Análisis de las Encuestas

El análisis global de las encuestas revela un escenario de informalidad predominante que sustenta la necesidad técnica para el desarrollo del proyecto. Al sintetizar los datos de arrendatarios y arrendadores, se identifican tres ejes críticos: fragilidad administrativa, desconexión financiera y estancamiento operativo.

Fragilidad Administrativa y Movilidad Estudiantil

Al analizar los resultados, surge una evidente vulnerabilidad jurídica originada porque el 64% de los arrendatarios vive bajo acuerdos verbales, realidad que corre en paralelo con el 72% de los arrendadores que utiliza cuadernos o registros manuales, agravándose esta desorganización por la alta movilidad, pues el 70% de los estudiantes reside en su inmueble actual por menos de un año, lo cual obliga a considerar la migración digital como el paso lógico para garantizar la integridad de la información sensible, logrando que el sistema propuesto funcione como un

repositorio flexible de archivos críticos, tales como matrículas o documentos de identidad, que brindan protección y respaldo a los involucrados.

Brecha en la Gestión Financiera

La situación económica actual genera incertidumbre ya que el 70% de los estudiantes paga sus cuotas mensuales en efectivo directamente al propietario. Esta preferencia por el dinero físico explica por qué el 60% de los inquilinos nunca recibe recibos de pago formales, dejando a 30 estudiantes sin respaldo legal en caso de posibles reclamaciones. Los propietarios también tienen dificultades con esto: el 60% de ellos cree que es necesario generar una lista de morosos para monitorear efectivamente las propiedades. Hay una actitud favorable hacia el cambio tecnológico, ya que el 84% de los inquilinos quiere recibir recordatorios automáticos y el 88% de los propietarios dice que esto les ahorrará tiempo valioso.

Eficiencia Operativa e Incidencias

El problema de los problemas técnicos sigue siendo un tema abierto, con el 50% de las reparaciones tardando más de 1 semana en resolverse. Esta demora es resultado de la comunicación informal, con el 60% de las quejas llegando a través de whatsApp o por teléfono sin un registro administrativo claro. La aceptación del módulo de tickets es enorme: el 90% de los estudiantes lo encuentra muy útil para documentar daños con fotos. En comparación, el 80% de los propietarios está dispuesto a actualizar el estado de las reparaciones (como "en proceso" o "resuelto" en la web) ellos mismos. La retroalimentación positiva y los indicadores de progreso asegurarán la trazabilidad técnica y el descuido del pasado se transferirá a una gestión profesional que se dirigirá hacia la comunidad universitaria en la parroquia de Guanujo.

4.1.3 Historias de usuarios

Los requerimientos definen las acciones específicas que el sistema debe ejecutar para satisfacer las necesidades de cada perfil de usuario. Pero en la metodología XP prácticamente las historias de usuario son los requerimientos funcionales. El cliente escribe historias breves en tarjetas que describen lo que el sistema debe hacer desde su perspectiva.

Tabla 22. Historia de Usuario Inicio de sesión y recuperación de contraseña

Historia de Usuario

Número: HU1		**Usuario:** Arrendador, Arrendatario, usuario de control parental.	**Stakeholder:** Arrendador, estudiante
Nombre de la historia:		Inicio de sesión y recuperación de contraseña	
Prioridad:	Alta	**Riesgo en desarrollo:** Baja	
Puntos estimados:	3 días	**Iteración asignada:** 1	
Programador responsable: Kevin Andrade P.			
Descripción: El sistema permite el acceso a los usuarios mediante la validación de sus credenciales: correo electrónico y contraseña. El acceso estará restringido según el rol asignado en la base de datos PostgreSQL. En caso de pérdida de contraseña el usurario podrá recuperarla usando el correo Gmail ingresado al momento del registro, el cual le llegará una notificación en el correo con un enlace para que pueda reestablecer la contraseña.			
Observaciones: El usuario ingresara con las credenciales de acceso que usó al momento del registro.			

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 23. Historia de Usuario Registro de Usuario

Historia de Usuario

Número: HU2		**Usuario:** Arrendador, Arrendatario, usuario de control parental.	**Stakeholder:** Arrendador, estudiante
Nombre de la historia:		Registro de Usuario	
Prioridad:	Alta	**Riesgo en desarrollo:** Media	
Puntos estimados:	4 días	**Iteración asignada:** 1	
Programador responsable: Kevin Andrade P.			
Descripción: El sistema permite registrar a los usuarios con sus datos personales y un rol que se asignen, para acceder a las funcionalidades correspondientes al perfil dentro de la plataforma.			
Observaciones: El proceso de registro se adapta según el rol de cada usuario, exigiendo que el arrendatario aporte el dato de su carrera universitaria y el rol del usuario de control parental también se le exige el parentesco, mientras todos los perfiles deben proporcionar su correo Gmail, clave de acceso, nombres, apellidos, residencia, número de cédula y edad, iniciando así un flujo donde el sistema dispara automáticamente un mensaje de confirmación hacia la dirección electrónica ingresada, activando la cuenta solo tras la verificación, cuidando siempre que las contraseñas se resguarden mediante algoritmos de encriptación y que el usuario reciba alertas tempranas al correo ante cualquier novedad relevante del entorno web.			

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 24. Historia de Usuario Registro de Inmueble

Historia de Usuario			
Número: HU3		**Usuario:** Arrendador	**Stakeholder:** Arrendador
Nombre de la historia:		Registro de Inmueble	
Prioridad:	Alta	**Riesgo en desarrollo:** Media	
Puntos estimados:	4 días	**Iteración asignada:** 2	
Programador responsable: Kevin Andrade P.			
Descripción: El sistema permite a los arrendadores registrar los inmuebles con toda su información detallada, para administrarlos de forma organizada y tenerlos disponibles para su vinculación con los arrendatarios.			
Observaciones: Solo el arrendador posee el permiso para registrar propiedades en la plataforma, gestionando cada inmueble de forma individual mediante la entrada de datos como características generales, medidas de espacio con número de cuartos, baños y metraje, dirección exacta, servicios básicos y detalles como focos o cortinas, siendo requisito indispensable subir al menos una fotografía para que el sistema valide el registro, vinculando automáticamente cada lugar con el perfil de quien lo registró.			

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 25. Historia de Usuario Vinculación de Usuarios con el Inmueble

Historia de Usuario		
Número: HU4	**Usuario:** Arrendador	**Stakeholder:** Arrendador

Nombre de la historia:		Vinculación de Usuarios con el Inmueble
Prioridad:	Alta	**Riesgo en desarrollo:** Media
Puntos estimados:	5 días	**Iteración asignada:** 2
Programador responsable: Kevin Andrade P.		
Descripción: El sistema permite al arrendador vincular a un arrendatario y a un usuario de control parental con un inmueble registrado, para formalizar digitalmente la relación de arriendo y dar acceso a los documentos y funcionalidades correspondientes.		
Observaciones: Para enlazar al arrendatario con el inmueble, el arrendador solamente necesita escribir el correo electrónico del inquilino, procedimiento idéntico al que aplica para asignar al usuario de control parental, obligando al dueño a adjuntar el contrato en PDF junto a las normas de convivencia durante este proceso, logrando que el sistema envíe al instante una notificación tanto al correo como al panel interno del inquilino, activando así la vinculación de forma automática para dejarlo marcado como habitante activo, mientras el software bloquea cualquier intento fallido si la dirección no existe mediante un mensaje de advertencia.		

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 26. Historia de Panel de Auditoría y Validación de Cobros

Historia de Usuario		
Número: HU5	**Usuario:** Arrendador	**Stakeholder:** Arrendador
Nombre de la historia:	Panel de Auditoría y Validación de Cobros	

Prioridad:	Alta	**Riesgo en desarrollo:** Media
Puntos estimados:	3 días	**Iteración asignada:** 2
Programador responsable: Kevin Andrade P.		
Descripción: El sistema permite visualizar en un panel el historial de pagos realizados por los inquilinos según cada inmueble, además de consultar los datos de los cobros, como fecha, tipo de pago y evidencia almacenada del pago, captura de transferencia o deposito.		
Observaciones: **Ninguna**		

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 27. Historia de Panel de Pagos y Justificativos Digitales

Historia de Usuario			
Número: HU6		**Usuario:** Arrendatario	**Stakeholder:** Arrendador
Nombre de la historia:		Panel de Pagos y Justificativos Digitales	
Prioridad:	Media	**Riesgo en desarrollo:** Baja	
Puntos estimados:	3 días	**Iteración asignada:** 2	
Programador responsable: Kevin Andrade P.			
Descripción:			

El sistema permite visualizar en un panel historial de pagos mensuales del inmueble que alquila, además de consultar los datos de los pagos, como fecha, tipo de pago y evidencia almacenada del pago, captura de transferencia o deposito.
Observaciones: **Ninguna**

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 28. Historia de Gestión de Contratos y Normas de Convivencia

Historia de Usuario			
Número: HU7		**Usuario:** Arrendador, Arrendatario	**Stakeholder:** Arrendador, estudiante
Nombre de la historia:		Gestión de Contratos y Normas de Convivencia	
Prioridad:	Alta	**Riesgo en desarrollo:** bajo	
Puntos estimados:	5 días	**Iteración asignada:** 1	
Programador responsable: Kevin Andrade P.			
Descripción: El sistema almacena contratos digitales y reglamentos generales de cada inmueble para su consulta inmediata. El arrendador sube los documentos y el estudiante los visualiza desde su perfil personal.			
Observaciones:			

Los archivos se cargan preferiblemente en formato PDF para asegurar su integridad. El estudiante debe marcar una casilla de "Ver" desplegando un modal donde se visualizan las reglas.

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 29. Historia de Control de Gastos

Historia de Usuario			
Número: HU8		**Usuario:** Arrendatario	**Stakeholder:** Arrendador
Nombre de la historia:		Control de Gastos	
Prioridad:	Alta	**Riesgo en desarrollo:** Baja	
Puntos estimados:	3 días	**Iteración asignada:3**	
Programador responsable: Kevin Andrade P.			
Descripción: El sistema registra los gastos de los servicios básicos y gastos por mantenimientos o reparaciones de los inmuebles.			
Observaciones: **Ninguna**			

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 30. Historia de Seguimiento de Cobros

Historia de Usuario		
Número: HU9	**Usuario:** Arrendador	**Stakeholder:** Arrendador

Nombre de la historia:		Seguimiento de Cobros
Prioridad:	Alta	**Riesgo en desarrollo:** Baja
Puntos estimados:	3 días	**Iteración asignada:** 3
Programador responsable: Kevin Andrade P.		
Descripción: El sistema notifica en un panel al arrendador los inquilinos que estén en mora o tengan pagos pendientes, así también como da aviso de aquellos inquilinos que ya está cerca su fecha de pago en un lapso de dos días a el pago del alquiler.		
Observaciones: Se generan alertas visuales de color rojo para identificar rápidamente a los arrendatarios morosos.		

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 31. Historia de Usuario Seguimiento de pagos

Historia de Usuario			
Número: HU10		**Usuario:** Arrendatario	**Stakeholder:** Estudiante
Nombre de la historia:		Seguimiento de pagos	
Prioridad:	alta	**Riesgo en desarrollo:** Baja	
Puntos estimados:	3 días	**Iteración asignada:** 3	
Programador responsable: Kevin Andrade P.			

Descripción: El sistema notifica al arrendatario cuando tengan pagos pendientes, así también como da aviso cuando ya está cerca su fecha de pago en un lapso de dos días a el pago del alquiler.
Observaciones: Se generan alertas visuales de color rojo para identificar rápidamente que pago está pendiente o que pago ya está cerca.

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 32. Historia de Gestión de Incidencias mediante Tickets

Historia de Usuario			
Número: HU11		**Usuario:** Arrendador	**Stakeholder:** Arrendador
Nombre de la historia:		Gestión de Incidencias mediante Tickets	
Prioridad:	Alta	**Riesgo en desarrollo:** Media	
Puntos estimados:	6 días	**Iteración asignada:** 3	
Programador responsable: Kevin Andrade P.			
Descripción: El sistema permite reportar incidencias en el inmueble adjuntando fotos y descripciones breves. El arrendador recibe la notificación, asigna una urgencia y actualiza el estado de la reparación hasta su cierre.			
Observaciones:			

El flujo de estados incluye: Pendiente, En Proceso y Resuelto. Cada ticket genera una evidencia trazable con fecha y hora exacta del reporte realizado por el estudiante de la UEB. **Ninguna**

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 33. Historia de Registro de Pagos con Comprobante Digital

Historia de Usuario			
Número: HU12		**Usuario:** Arrendatario	**Stakeholder:** Estudiante
Nombre de la historia:		Registro de Pagos con Comprobante Digital	
Prioridad:	Media	**Riesgo en desarrollo:** Baja	
Puntos estimados:	5 días	**Iteración asignada: 3**	
Programador responsable: Kevin Andrade P.			
Descripción: El sistema registra los pagos mensuales mediante la carga de capturas de transferencia o depósitos bancarios. El estudiante mantiene el control histórico de sus justificativos financieros dentro de la plataforma.			
Observaciones: El arrendador debe validar la captura para que el pago figure como "Aprobado" en el estado de cuenta. El sistema impide subir archivos que no sean imágenes o documentos PDF.			

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 34. Historia de Supervisión de Estado Financiero y Residencial

Historia de Usuario			
Número: HU13		**Usuario:** Control Parental	**Stakeholder:** Arrendador
Nombre de la historia:		Supervisión de Estado Financiero y Residencial	
Prioridad:	Alta	**Riesgo en desarrollo:** bajo	
Puntos estimados:	4 días	**Iteración asignada:** 4	
Programador responsable: Kevin Andrade P.			
Descripción: El sistema muestra al representante legal el historial de transacciones financieras del estudiante. El usuario revisa también el progreso de los reportes de incidencias para asegurar condiciones adecuadas del inmueble.			
Observaciones: Este rol tiene permisos restringidos únicamente a la visualización. No puede editar pagos, eliminar contratos ni alterar el flujo de los tickets de mantenimiento.			

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 35. Historia de Generación de Reportes PDF

Historia de Usuario		
Número: HU14	**Usuario:** Arrendador, Arrendatario	**Stakeholder:** Arrendador, estudiante
Nombre de la historia:	Generación de Reportes PDF	

Prioridad:	Alta	**Riesgo en desarrollo:** Baja
Puntos estimados:	5 días	**Iteración asignada:** 4
Programador responsable: Kevin Andrade P.		
Descripción: El sistema genera automáticamente reportes de pagos e historiales individuales en formato PDF descargable. Los documentos sirven como respaldo formal ante cualquier auditoría o conflicto administrativo.		
Observaciones: Ninguna		

Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 36. Historia de Configuración de Seguridad y Accesos

Historia de Usuario			
Número: HU15		**Usuario:** Arrendador, Arrendatario	**Stakeholder:** Arrendador, estudiante
Nombre de la historia:		Configuración de Seguridad y Accesos	
Prioridad:	Alta	**Riesgo en desarrollo:** Baja	
Puntos estimados:	3 días	**Iteración asignada:** 1	
Programador responsable: Kevin Andrade P.			
Descripción:			

El sistema permite configurar niveles de acceso y asegurar la estabilidad global de la base de datos PostgreSQL. El técnico gestiona los permisos para proteger la información sensible de toda la comunidad universitaria.
Observaciones: Ninguna

Nota. Elaborado por: Andrade P. Kevin (2026).

4.1.4 Requerimientos no Funcionales

Estos requerimientos garantizan que la plataforma funcione bajo estándares de seguridad, estabilidad y confianza del sistema web.

Tabla 37. Requerimientos no funcionales

Código	Nombre	Descripción	Medición	Justificación	Prioridad
RNF01	**Rendimiento**	El sistema carga reportes financieros y de incidencias en menos de 3 segundos, optimizando las consultas mediante el motor PostgreSQL.	Tiempo de respuesta del servidor bajo carga nominal.	La eficiencia operativa evita demoras administrativas frustrantes para los usuarios en Guanujo.	**Alta**
RNF02	**Compatibilidad**	La plataforma web funciona correctamente en navegadores	Pruebas de renderizado multiplataforma sin errores visuales.	Los estudiantes de la UEB acceden al sistema desde diversos	**Alta**

		modernos y dispositivos móviles, adaptando su interfaz de forma totalmente responsiva.		dispositivos portátiles constantemente.	
RNF03	**Escalabilidad**	La arquitectura soporta un crecimiento del 100% en el volumen de datos de alquileres sin degradar la velocidad.	Pruebas de carga.	El sistema debe crecer junto a la demanda habitacional proyectada para el año 2026.	**Alta**
RNF04	**Seguridad**	Los registros de pagos y documentos sensibles permanecen cifrados, protegiendo la integridad financiera de los involucrados.	Auditoría de protocolos SSL/TLS y niveles de acceso mediante herramientas dedicadas como ZAP (Zed Attack Proxy)	La integridad de datos protege legalmente al arrendador y al inquilino estudiantil.	**Muy Alta**

RNF05	**Usabilidad**	La interfaz resulta intuitiva para propietarios con poca experiencia tecnológica, facilitando una navegación sencilla y directa.	Pruebas de usabilidad con usuarios reales.	La facilidad de uso garantiza que los arrendadores abandonen sus registros en cuadernos físicos.	**Alta**
RNF06	**Seguridad de Autentificación**	El acceso requiere credenciales únicas verificadas, impidiendo que personas no autorizadas manipulen la información residencial almacenada.	Validación de tokens de sesión y encriptación de claves.	Solo usuarios válidos deben gestionar la información sensible de la universidad en la plataforma.	**Muy Alta**

Nota. Elaborado por: Andrade P. Kevin (2026).

4.1.5 Plan de Entregas

En esta etapa se organizan las historias de usuario previamente definidas, con el fin de establecer un plan de entregas coherente con los principios de la metodología XP. A partir de su análisis, se determinan las historias que tienen más prioridad y se agrupan en iteraciones que

permitan desarrollar funcionalidades de manera incremental, asegurando avances continuos y medibles en cada ciclo. Cabe considerar que el desarrollo del proyecto se realiza con recursos limitados, específicamente con un solo programador, lo que influye en la estimación de tiempos y en la planificación de las entregas. Sin embargo, este escenario se ajusta al enfoque flexible de XP, permitiendo adaptar el ritmo de trabajo y realizar ajustes conforme evoluciona el proyecto, sin perder de vista los objetivos planteados.

Tabla 38. Tabla para la estimación de entregas por cada historia de usuario

Módulo Usuario	**Historia de Usuario**	**Tiempo Estimado (Días)**	**Iteraciones**			
			1	**2**	**3**	**4**
Arrendador, Arrendatario, Superadmin, Control Parental	**HU01** Inicio de sesión y recuperación de contraseñas	3	**X**			
Arrendador, Arrendatario, Control Parental	**HU02** Registro de Usuario	4	**X**			
Arrendador, Arrendatario	**HU15** Configuración de Seguridad y Accesos	3	**X**			
Arrendador, Arrendatario	**HU07** Gestión de Contratos y Normas	5	**X**			
Arrendador	**HU03** Registro de Inmueble	4		**X**		
Arrendador	**HU04** Vinculación de Usuarios con el Inmueble	5		**X**		
Arrendador	**HU05** Panel de Auditoría y Validación de Cobros	3		**X**		

Arrendatario	**HU06** Panel de Pagos y Justificativos Digitales	3		**X**		
Arrendatario	**HU12** Registro de Pagos con Comprobante Digital	5			**X**	
Arrendatario	**HU08** Control de Gastos	3			**X**	
Arrendador	**HU09** Seguimiento de Cobros (Alertas Mora)	3			**X**	
Arrendatario	**HU10** Seguimiento de pagos (Alertas vencimiento)	3			**X**	
Arrendador, Arrendatario	**HU11** Gestión de Incidencias (Tickets)	6			**X**	
Control Parental	**HU13** Supervisión de Estado Financiero y Residencial	4				**X**
Arrendador, Arrendatario	**HU14** Generación de Reportes PDF	5				**X**

Nota. Elaborado por: Andrade P. Kevin (2026).

4.2 Fase II: Diseño

4.2.1 Modelo de Vistas Arquitectónicas

Describir cómo está construido un sistema de software no es tarea fácil, porque distintas personas necesitan entender cosas distintas del mismo sistema. Un desarrollador quiere saber cómo está organizado el código, mientras que alguien de infraestructura necesita saber en qué servidor corre cada componente.

Para resolver ese problema, Kruchten (1995) propone el Modelo 4+1, que organiza la descripción de una arquitectura de software en cinco vistas complementarias: la vista lógica

muestra cómo se estructuran los objetos y datos del sistema, la vista de procesos explica cómo conviven y se sincronizan las tareas en ejecución, la vista física refleja cómo el software se distribuye sobre el hardware, y la vista de desarrollo describe cómo está organizado el código durante su construcción. Estas cuatro vistas se conectan mediante una quinta, la vista de escenarios, que usa casos de uso concretos para validar y explicar las decisiones tomadas en el sistema.

Aplicar este modelo al presente proyecto permite documentar el sistema desde múltiples ángulos, garantizando que cada aspecto técnico quede representado de forma clara y comprensible para todos los involucrados.

Tabla 39. El Modelo de "4+1" Vistas de la Arquitectura del Software

Vista	Objetos que usa	Objetivo principal de la vista
Vista Lógica	Clases, paquetes, objetos y asociaciones.	Mostrar la funcionalidad que el sistema ofrece al usuario. Se centra en "qué" hace el sistema y cómo se organiza internamente la lógica.
Vista de Procesos	Procesos, hilos (threads) y tiempos de ejecución.	Explicar cómo se comporta el sistema en tiempo real. Se enfoca en el rendimiento, la escalabilidad y cómo se comunican las partes mientras corren.
Vista de Desarrollo	Componentes, módulos, librerías y subsistemas.	Organizar el software desde el punto de vista del programador. Sirve para gestionar la gestión de archivos, versiones y la estructura del código.
Vista Física	Nodos, servidores, hardware y conexiones de red.	Definir cómo se instala el software en el mundo real. Muestra en qué servidores vivirá el código y cómo se conectan físicamente los equipos.
Escenarios (Casos de Uso)	Actores, casos de uso y secuencias de interacción.	Validar que todas las demás vistas funcionen juntas. Es el pegamento que asegura que el diseño realmente soluciona lo que el usuario necesita.

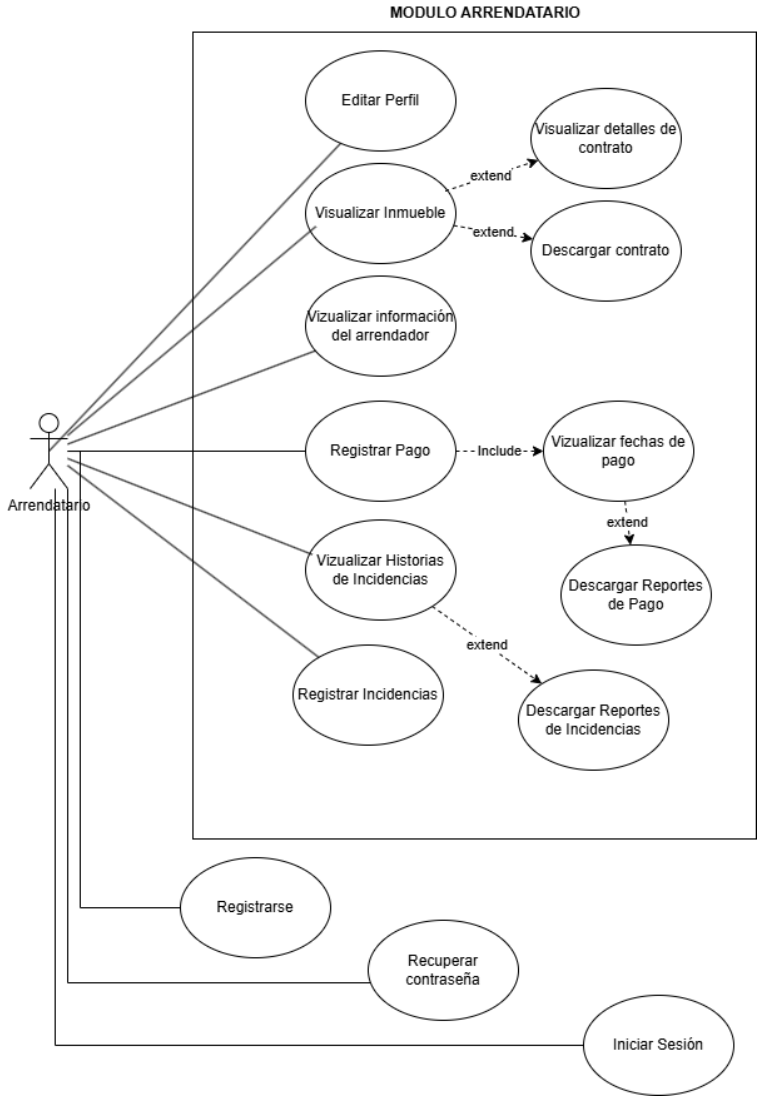
Nota. Elaborado por: Andrade P. Kevin (2026).

Vista de Escenarios

En esta vista se utilizan los casos de uso para representar las funcionalidades principales del sistema con sus actores involucrados. Es, en esencia, el corazón del modelo arquitectónico, ya que valida que el sistema haga exactamente lo que los usuarios necesitan. A través de los diagramas de casos de uso se puede visualizar de forma clara cómo interactúan los diferentes perfiles (arrendatario, arrendador, y el usuario de control parental) con el sistema " GestInWEB ", detallando las acciones específicas que cada uno puede ejecutar.

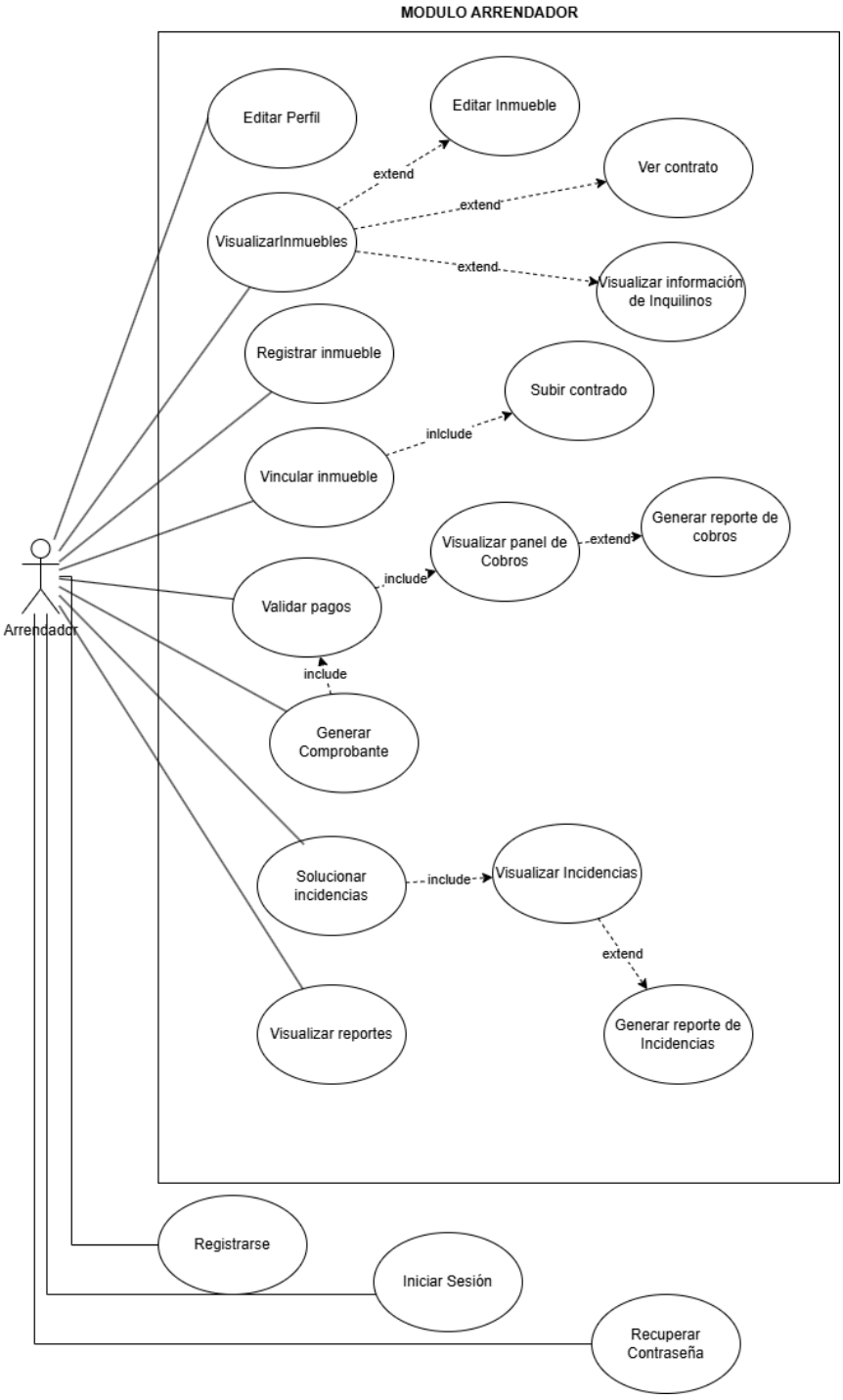
En esta sección se presentan los cuatro módulos principales que componen la solución. Cada diagrama ilustra el flujo de tareas y responsabilidades de los actores involucrados, garantizando que los requerimientos funcionales queden alineados con el uso real del sistema en el entorno de alquileres.

Figura 21. Diagrama de casos de usos del módulo del Arrendatario



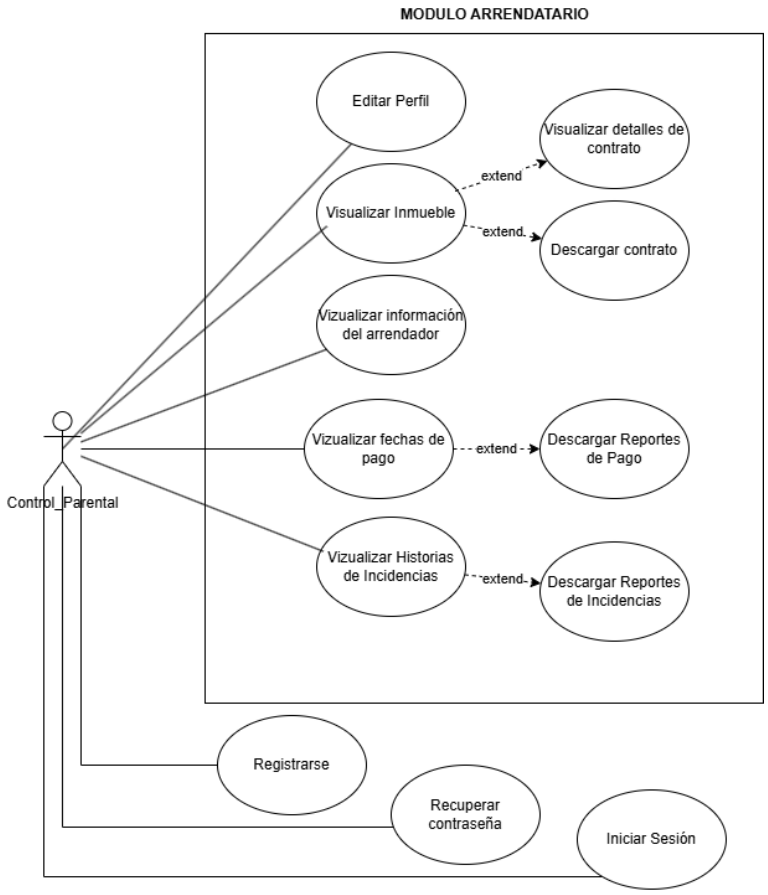
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 22. Diagrama de casos de usos del modulo arrendador



Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 23. Diagrama de casos de usos del modulo de control parental



Nota. Elaborado por: Andrade P. Kevin (2026).

Vista de Desarrollo

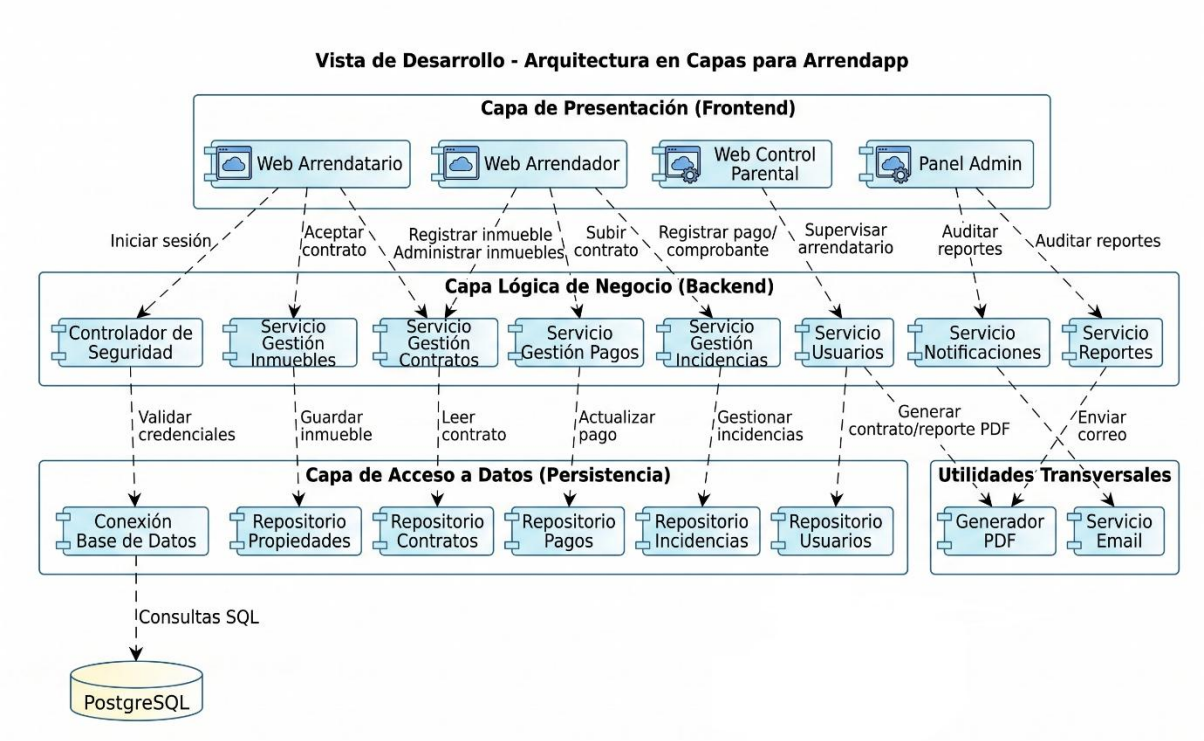
Esta vista estructura el sistema bajo un modelo de capas que separar las responsabilidades técnicas de las reglas de negocio, permitiendo que cada componente trabaje de forma independiente pero coordinada.

En la parte superior se encuentra la Capa de Presentación (Frontend) donde se organizan los módulos que gestionan la interacción directa con el usuario. En esta capa, los inquilinos, propietarios y usuarios de control parental están asociados con interfaces específicas. Estas interfaces son responsables de recopilar datos y mostrar información, por lo que todo el procesamiento real se deja a las capas intermedia e inferior para mantener la experiencia del usuario lo más sencilla posible.

La Capa de Lógica de Negocio (Backend) es el motor de la aplicación y procesa las tareas esenciales de gestión de contratos, registro de incidentes y verificación de pagos. Para complementar estos servicios, el sistema cuenta con un paquete de servicios transversales que brindan soporte técnico que no pertenece a la lógica propia del alquiler y que se puede realizar automáticamente, como la creación de informes PDF para contratos y la notificación por correo electrónico.

La base del sistema es la capa de acceso a datos. Gestiona la persistencia de la información de manera segura, aislando las consultas y conectando directamente el motor de la base de datos. La estructura de esta capa permite que los cambios en la base de datos no afecten a las otras capas.

Figura 24. Diagrama de Paquetes del sistema



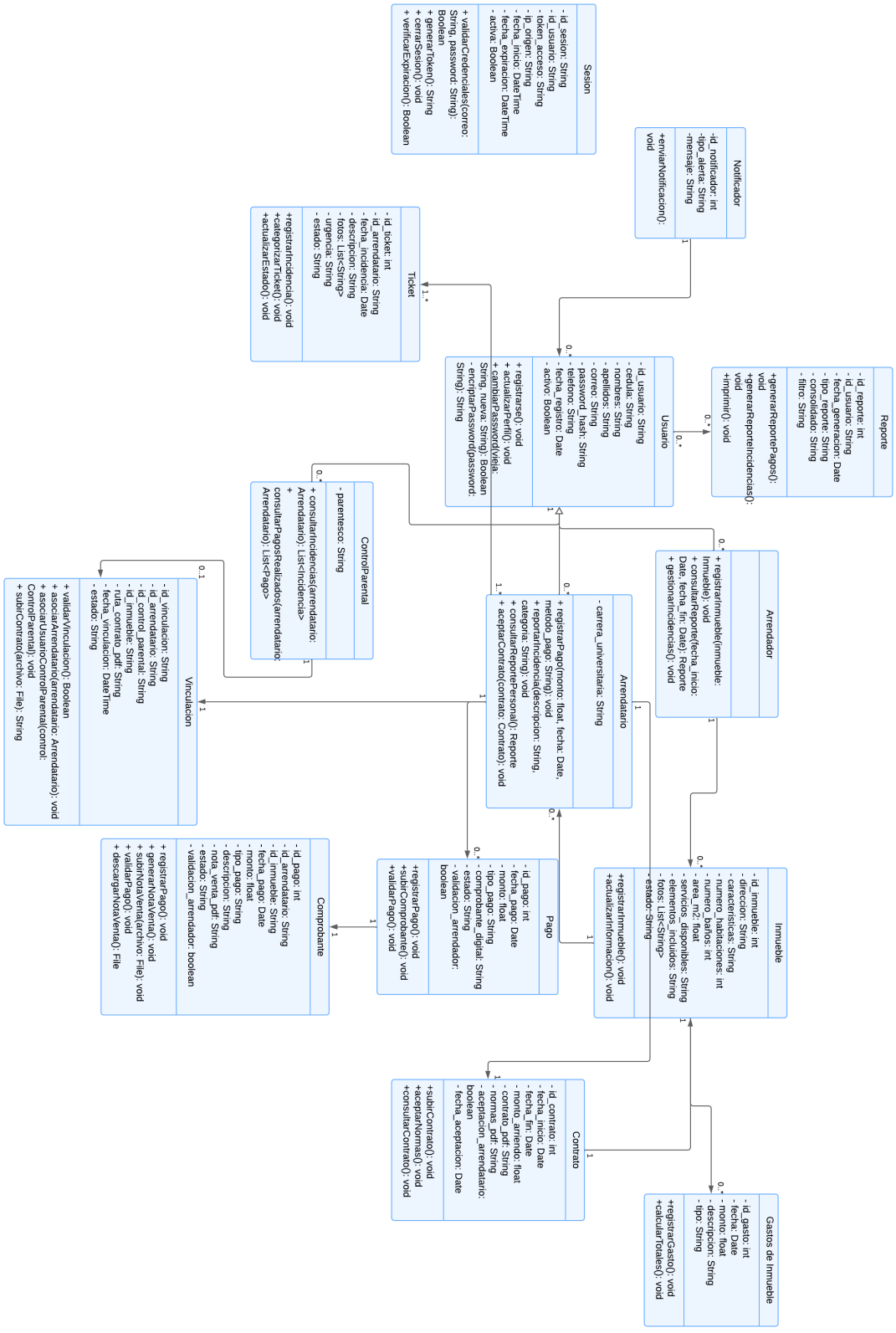
Nota. Elaborado por: Andrade P. Kevin (2026).

Vista Lógica

Mientras que la vista de escenarios nos muestra qué hace el usuario, la Vista Lógica se adentra en el "esqueleto" funcional de GestInWEB. Aquí es donde definimos la estructura interna del sistema, organizando la información en objetos y categorías lógicas que permiten que todo funcione de manera coherente.

En esta sección, el diagrama de clases detalla cómo se relacionan las entidades principales (como Inmuebles, Contratos, Pagos y Usuarios) y qué atributos o acciones posee cada una. Es, en esencia, el plano que explica cómo se traduce el negocio del alquiler a un lenguaje que el software puede procesar, asegurando que los datos fluyan correctamente entre los diferentes módulos del sistema.

Figura 25. Diagrama de Clases del sistema



Nota. Elaborado por: Andrade P. Kevin (2026).

Vista de Procesos

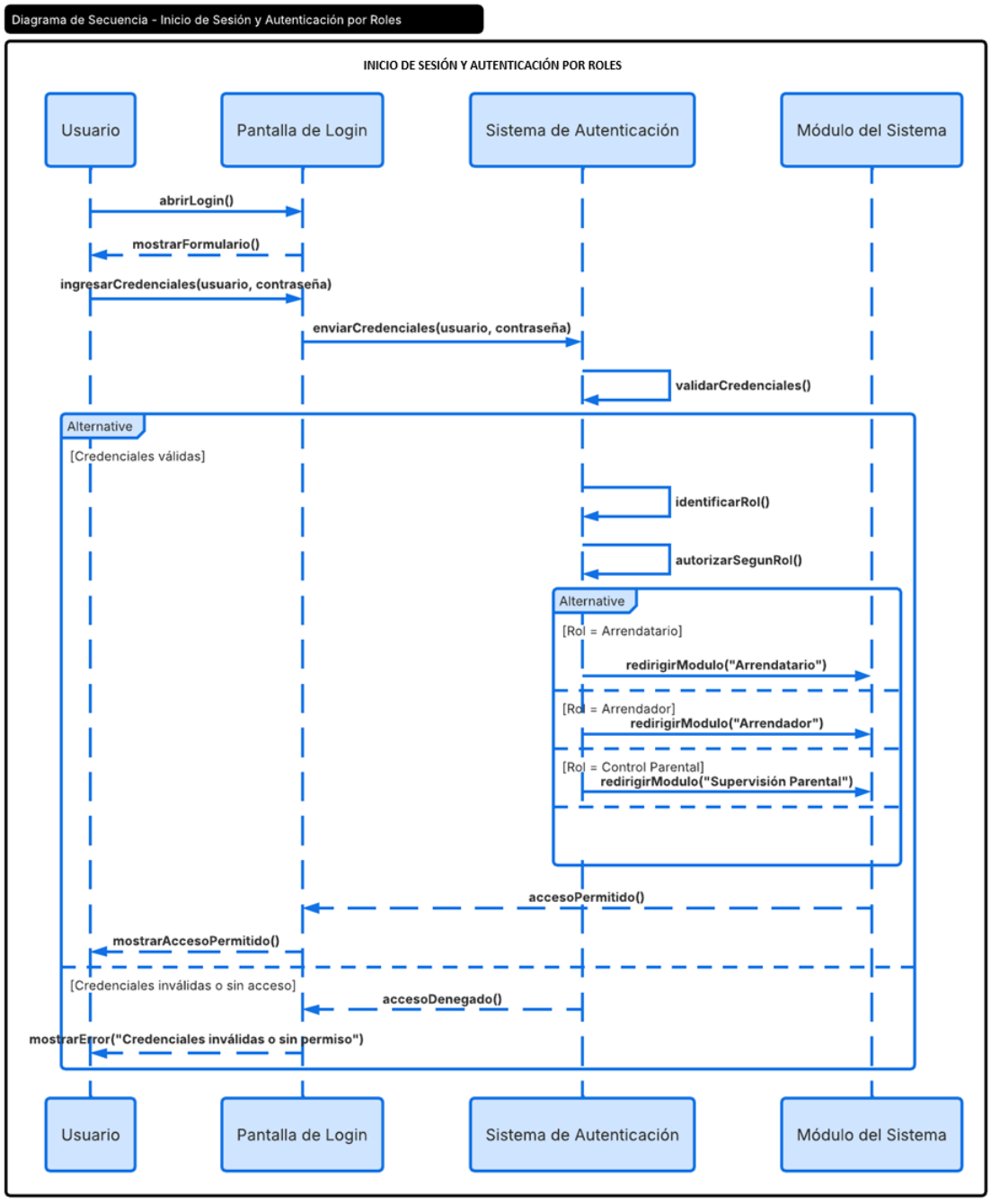
La Vista de Procesos se enfoca en el comportamiento dinámico de sistema, es decir, en cómo se comunican las distintas partes del software en tiempo real para completar una tarea específica.

Esta vista se ilustra a través de los diagramas de secuencia, mostrando el flujo de mensajes y la sincronización entre los objetos a lo largo del tiempo. Permitiendo asegurar que procesos críticos como la validación de un pago o la generación de un contrato ocurran en el orden correcto, evitando los famosos cuellos de botella y de esta manera se garantiza que el sistema responda de manera eficiente y segura ante las peticiones de los usuarios.

Vamos a centrarnos en las funcionalidades principales del sistema:

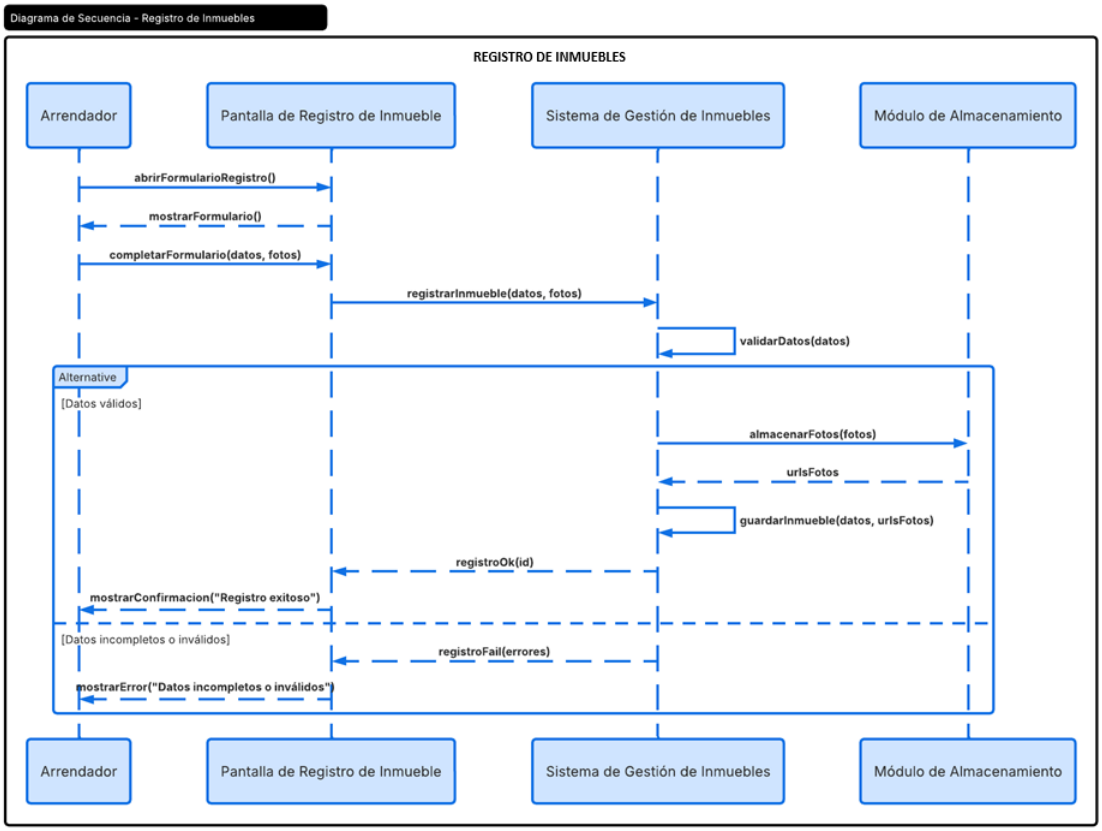
- Inicio de sesión
- Registro de inmuebles
- Vinculación de inmuebles
- Pagos y cobros
- Generar reportes
- Incidencias

Figura 26. Diagrama de secuencia de Inicio de sesión



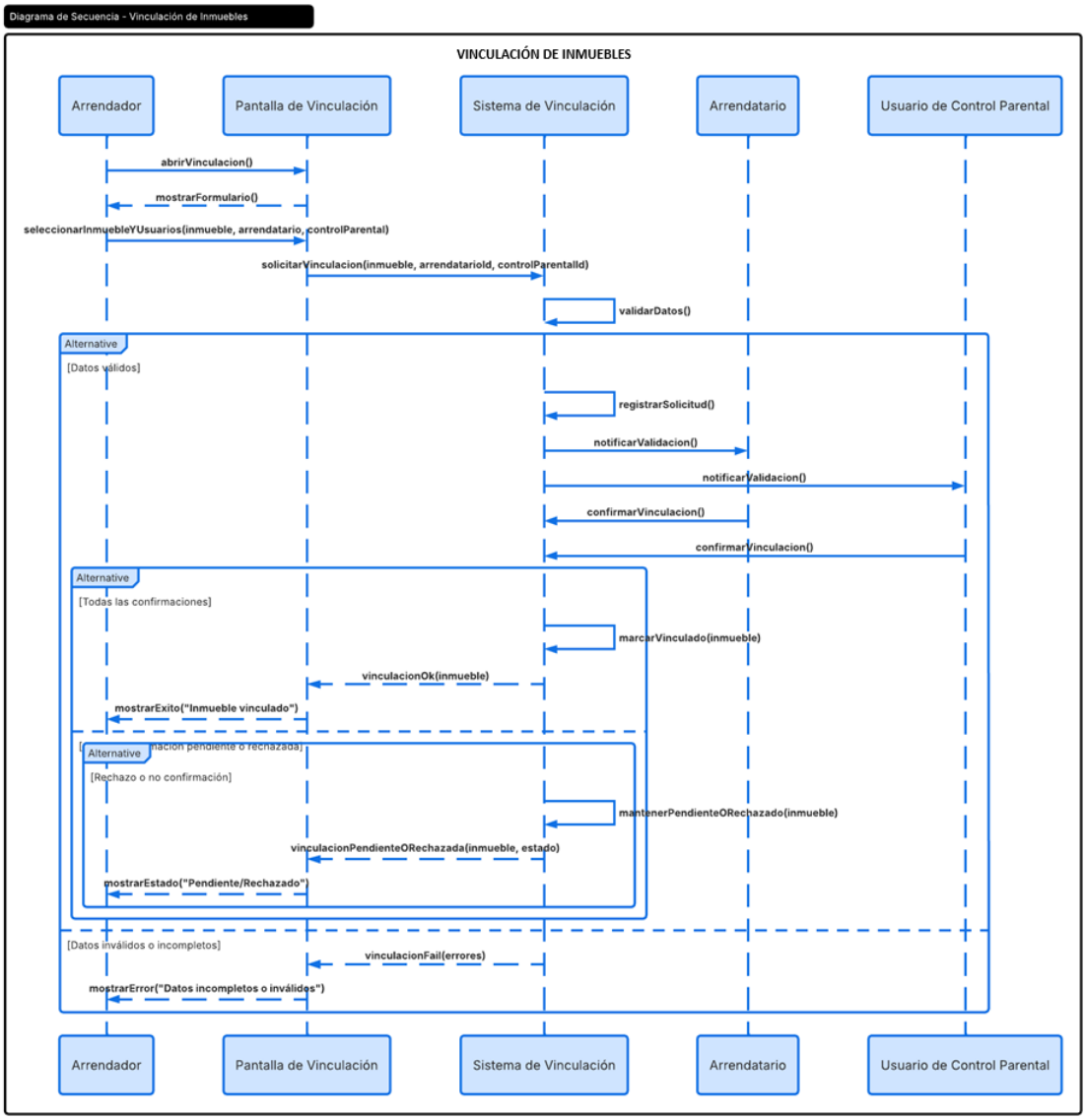
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 27. Diagrama de secuencia del Registro de inmuebles



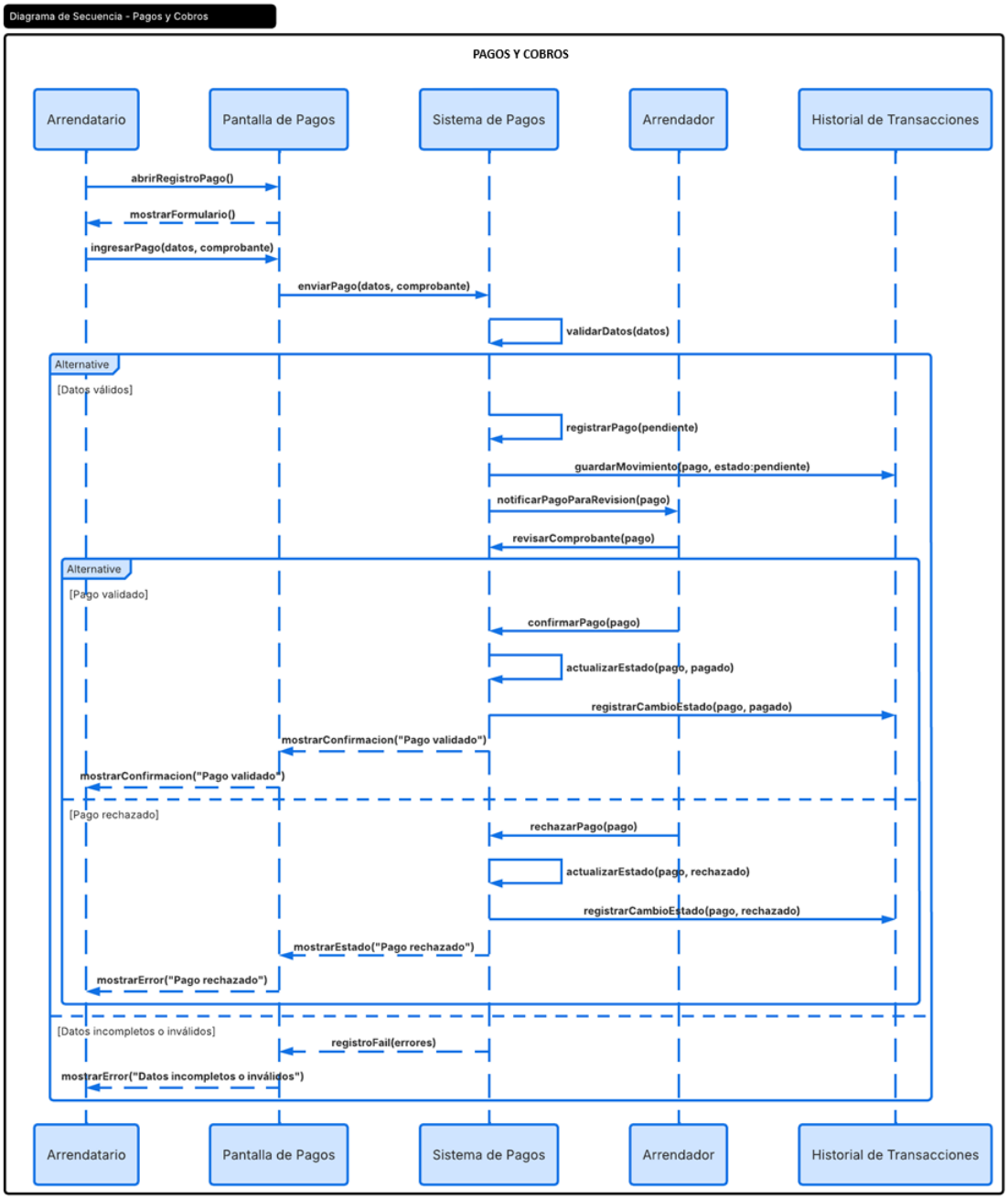
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 28. Diagrama de secuencia de Vinculación de inmuebles



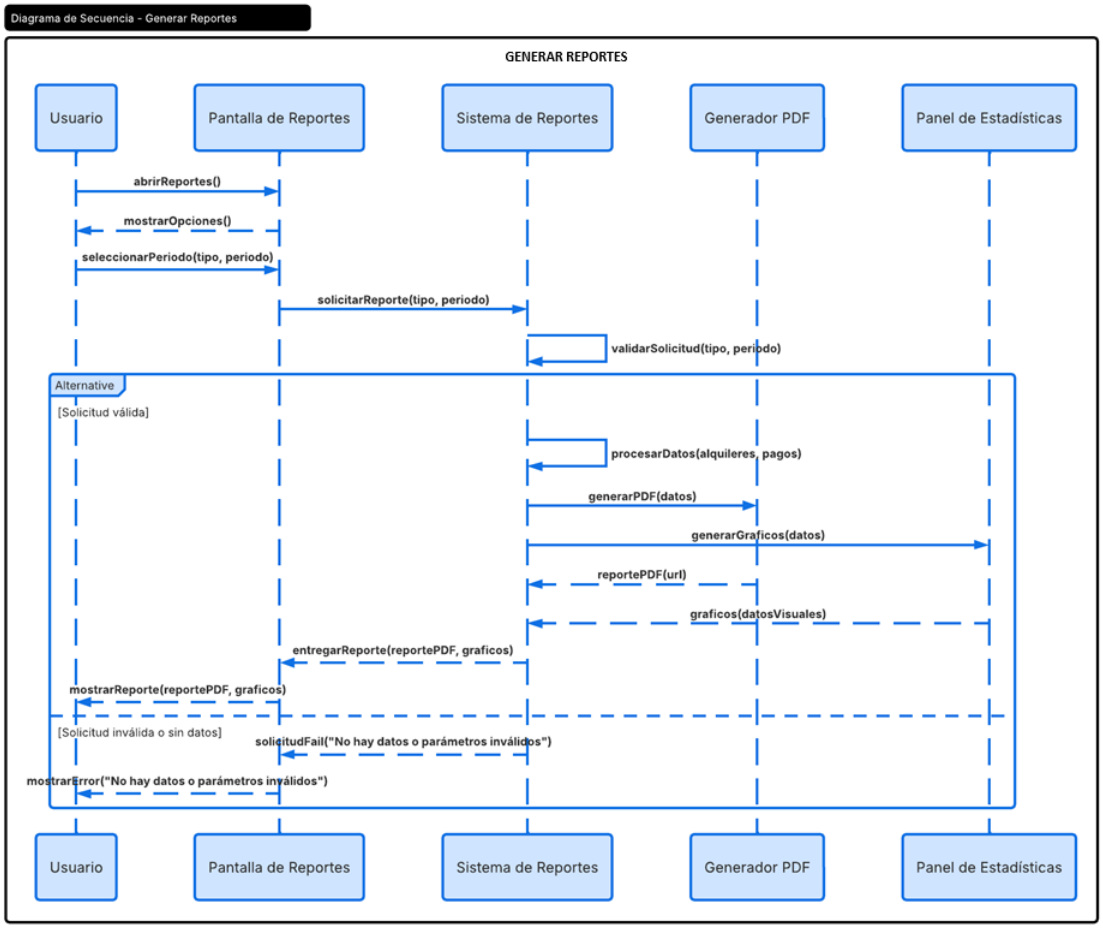
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 29. Diagrama de secuencia de Pagos y cobros



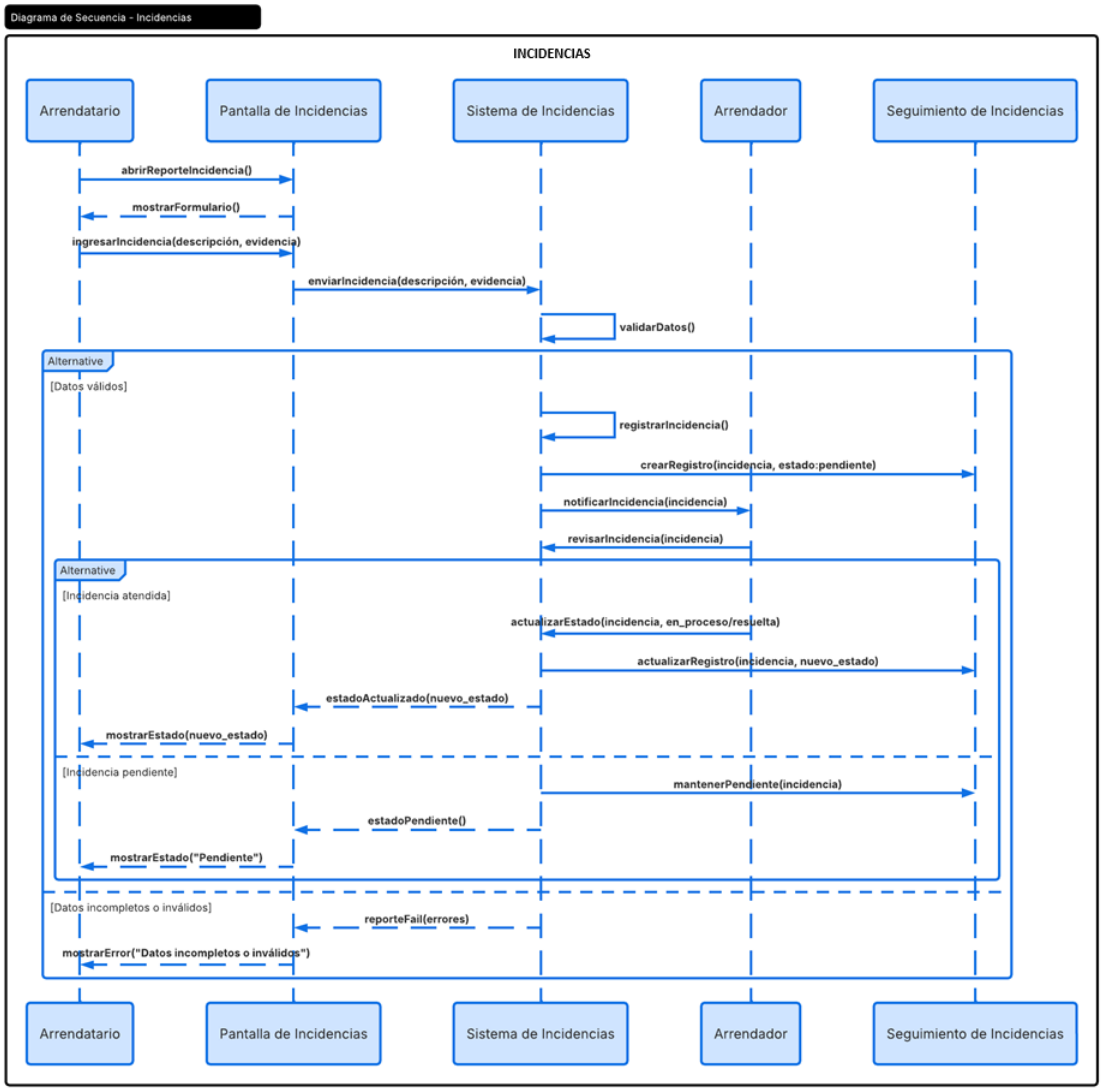
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 30. Diagrama de secuencia de Generar reportes



Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 31. Diagrama de secuencia de las Incidencias

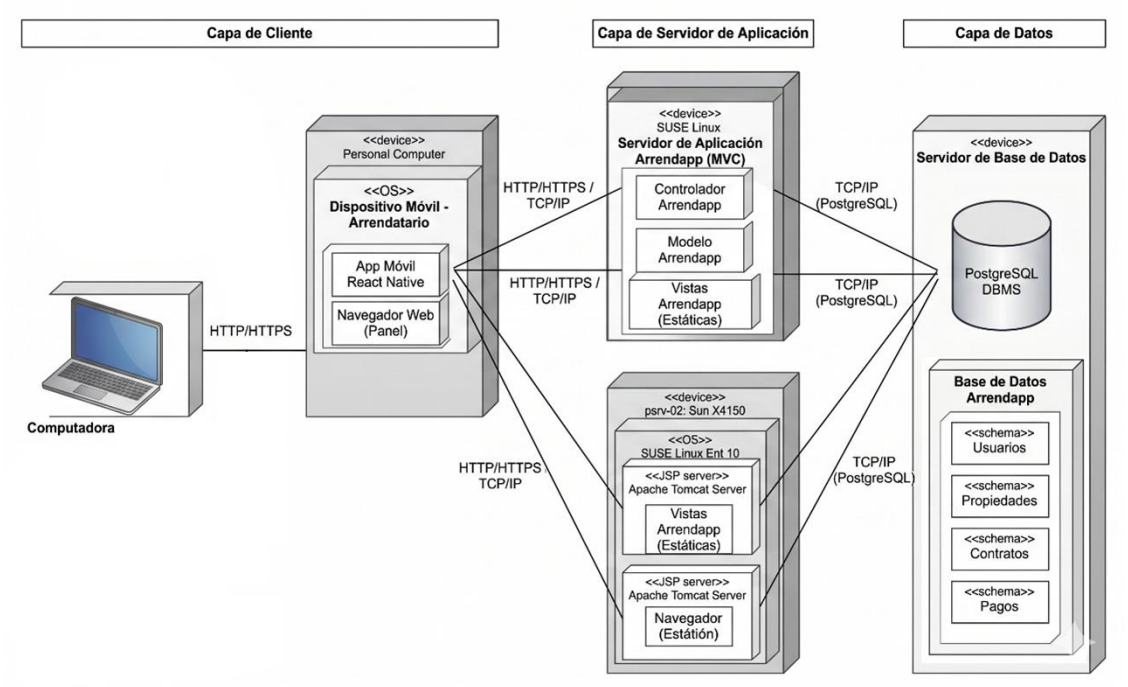


Nota. Elaborado por: Andrade P. Kevin (2026).

Vista de Física

La Vista Física se detalla cómo se distribuirá el sistema en la infraestructura de hardware y red a través del diagrama de despliegue. En esta vista se visualiza la ubicación de los servidores, los dispositivos de los usuarios y las conexiones físicas que los unen, lo cual es crucial para planificar la instalación, anticipar necesidades de ancho de banda y garantizar la escalabilidad y disponibilidad del sistema en un entorno de producción real, asegurando que el sistema funcione de manera robusta y eficiente para todos los involucrados.

Figura 32. Diagrama de despliegue del sistema



Nota. Elaborado por: Andrade P. Kevin (2026).

4.2.2 Patrón Arquitectónico

El sistema GestInWEB se ha construido bajo el patrón arquitectónico Modelo-Vista-Controlador (MVC), este patrón permite separar la lógica de negocio de la interfaz visual para asegurar un mantenimiento mucho más sencillo. Esta estructura es la que permite que, por ejemplo, un arrendador pueda registrar un inmueble en la pantalla sin que el código de la interfaz se mezcle con las reglas de validación de la base de datos.

Modelo

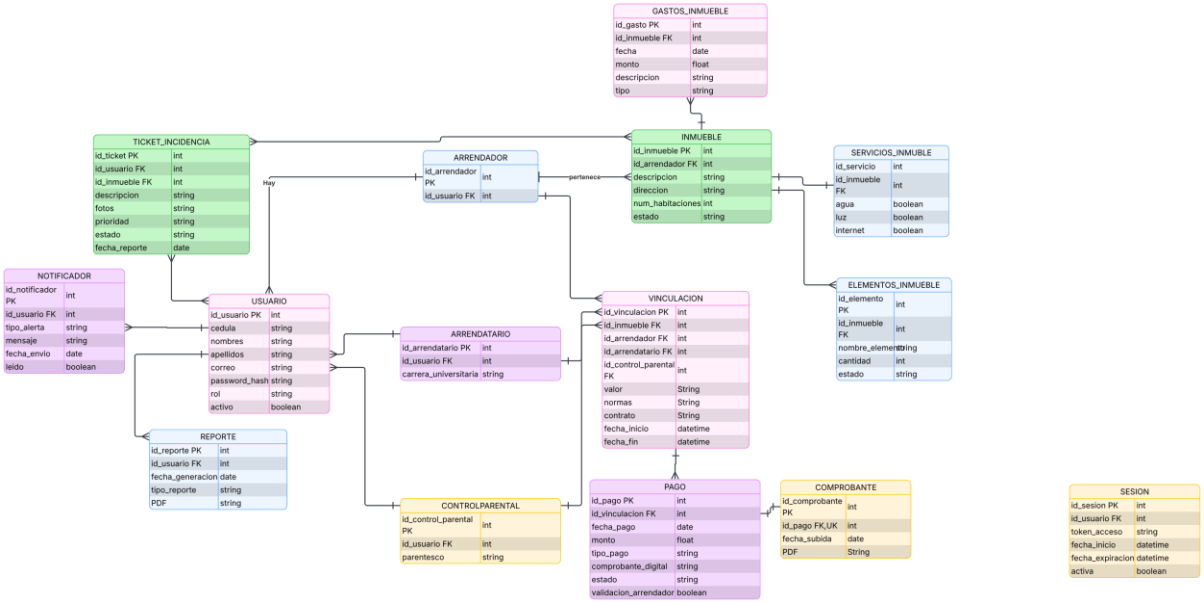
Es la representación de los datos y las reglas del sistema, donde gestionamos las entidades de inquilinos, propietarios y contratos en nuestra base de datos PostgreSQL. Aquí es donde vive la "verdad" de la información, asegurando que un pago solo se registre si cumple con los requisitos definidos en el backend.

Modelo Entidad Relación de la Base de datos

Si bien las vistas arquitectónicas nos explican cómo funciona el software y dónde se instala, el Modelo Entidad Relación se encarga de definir cómo se guarda la memoria de GestInWEB en la base de datos. Aquí es donde estructuramos la información de forma lógica y segura para que los datos no se pierdan y sean fáciles de consultar.

A través del Modelo Entidad-Relación (ER), representamos las tablas y las conexiones que permiten que el sistema "entienda" quién es quién. Por ejemplo, cómo un pago específico se vincula a un contrato, y ese contrato a un inmueble y a un inquilino determinado. Esta vista es el cimiento de la persistencia de datos en PostgreSQL, asegurando que toda la gestión administrativa y financiera del proyecto sea íntegra, organizada y escalable a largo plazo.

Figura 33. Diagrama Entidad-Relación del sistema



Nota. Elaborado por: Andrade P. Kevin (2026).

Vista

Se refiere a la parte visual y reactiva con la que interactúan los cuatro perfiles de usuario, desde el panel del arrendatario hasta la interfaz del usuario de control parental. Al estar separada del resto, podemos mejorar el diseño o la experiencia de usuario en React sin tener que reprogramar cómo se guardan los datos internamente.

Diseño de la Interfaz

Para el diseño de la interfaz se realizó una estructura visualmente atractiva y funcional de cada pantalla del sistema, tomando como base directa las historias de usuario que se realizó en la fase anterior. Se priorizó una navegación intuitiva, con rutas claras fácil de comprender para el usuario, ya que tienen niveles distintos de experiencia tecnológica logrando que el resultado sea un diseño limpio, responsivo y organizado por módulos, pensado para que cualquier usuario pueda operar el sistema sin necesidad de capacitación previa.

Figura 34. Logo del sistema



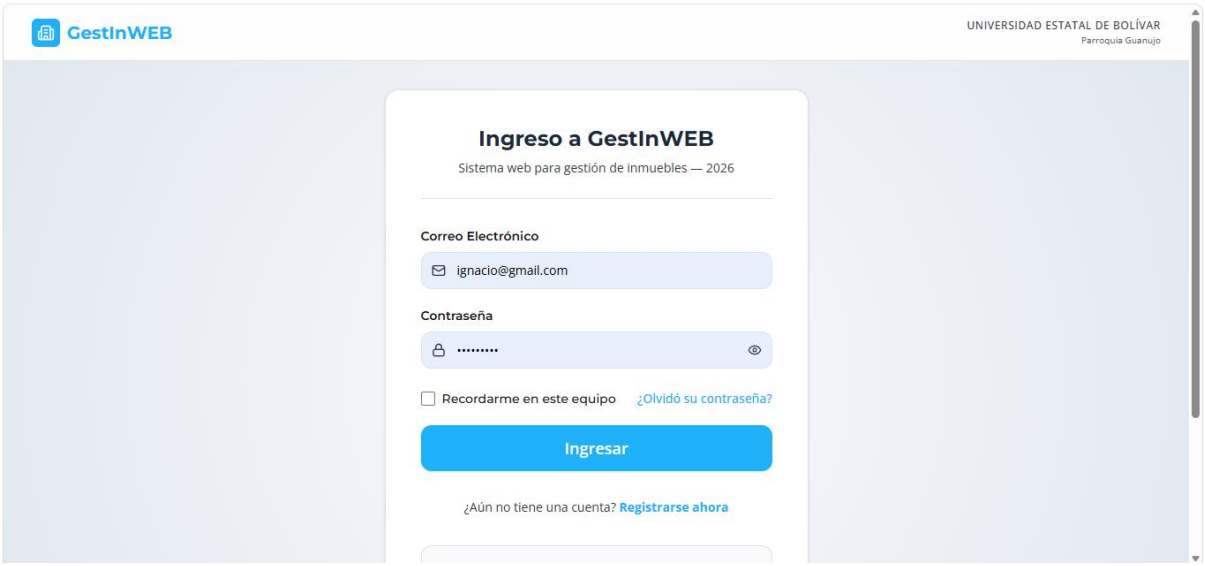
Nota. Elaborado por: Andrade P. Kevin (2026).

Tabla 40. Tabla de colores para el sistema GestInWEB

Clase	Descripción	Valor Hexadecimal
bg-sky-500	Azul principal (Botones y selección activa)	#0ea5e9
bg-white	Fondo de contenedores y campos de texto	#ffffff
bg-gray-50	Fondo general de la página (Gris muy claro)	#f9fafb
text-gray-900	Texto principal y títulos	#111827
text-gray-500	Texto secundario y placeholders	#6b7280
border-gray-200	Bordes de botones inactivos y divisores	#e5e7eb

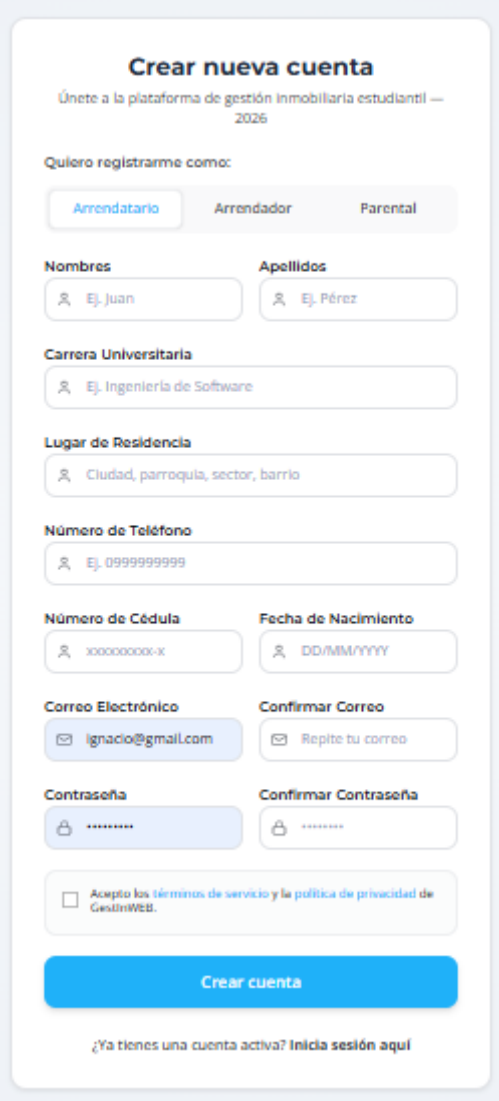
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 35. Interfaz de Inicio de sesión



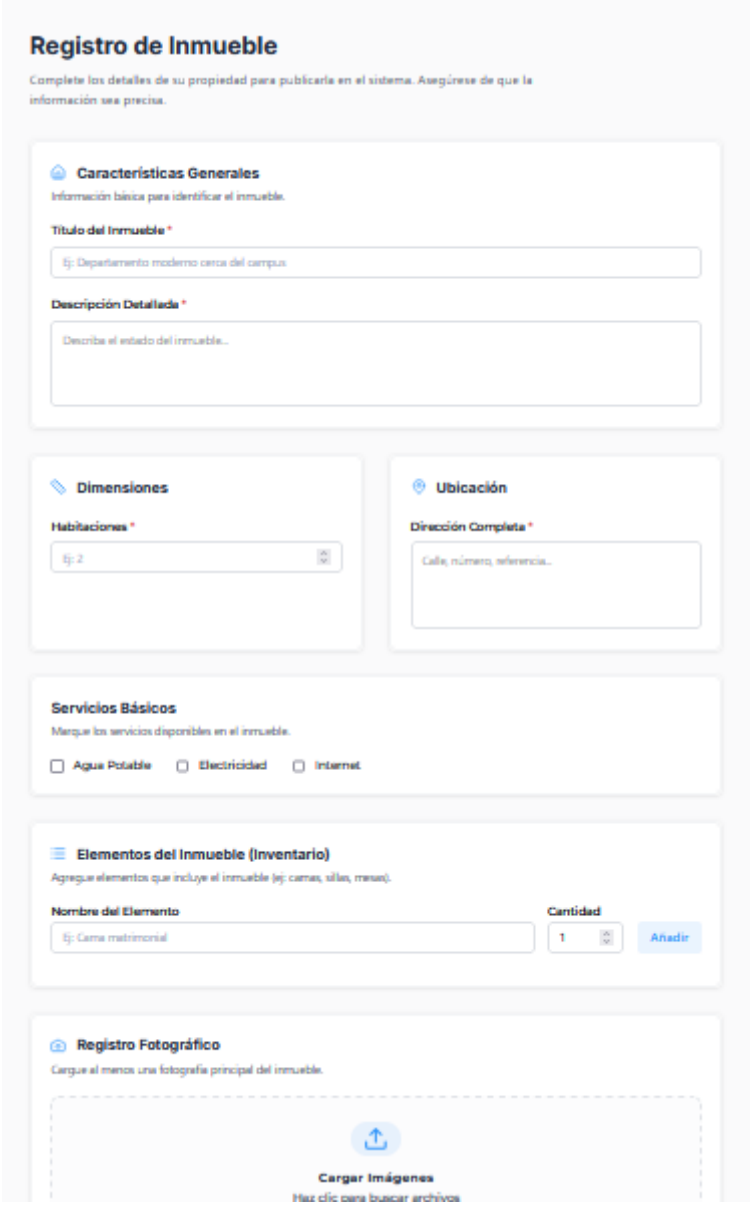
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 36. Interfaz de registro



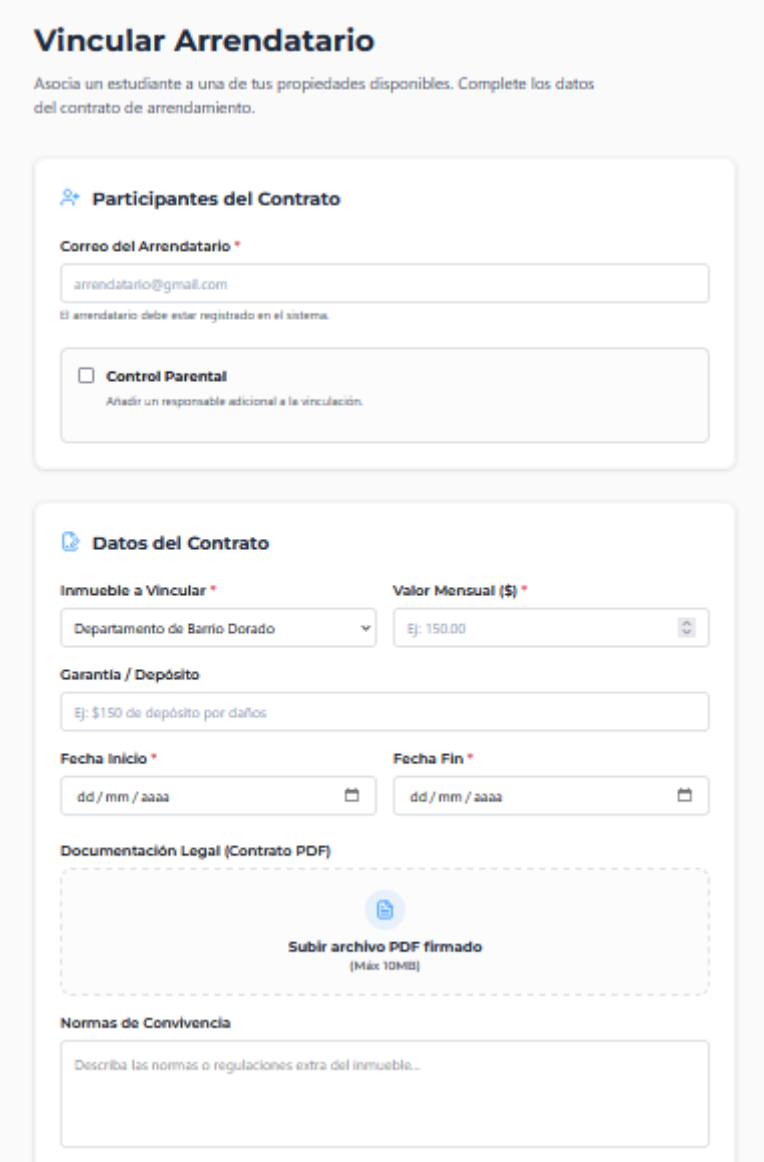
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 37. Interfaz de registro de inmueble



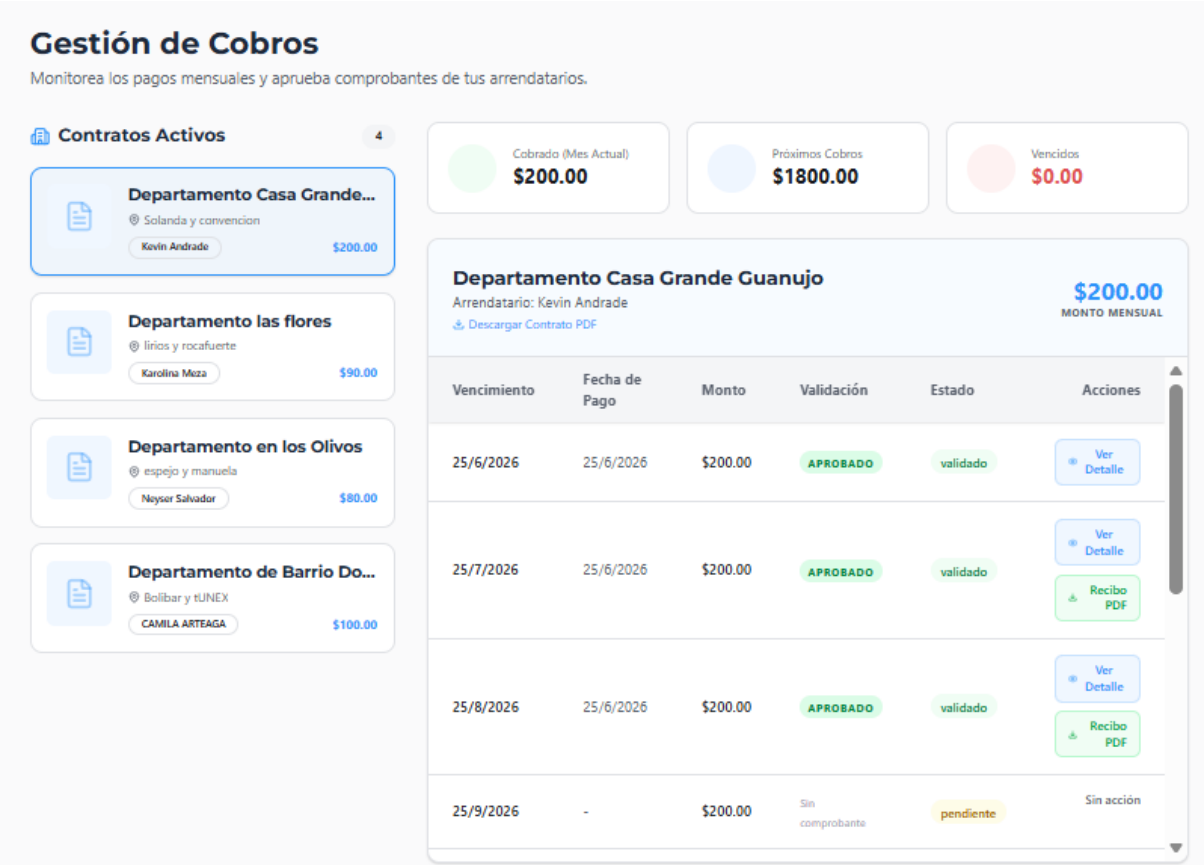
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 38. Interfaz de vinculación de usuarios con el inmueble



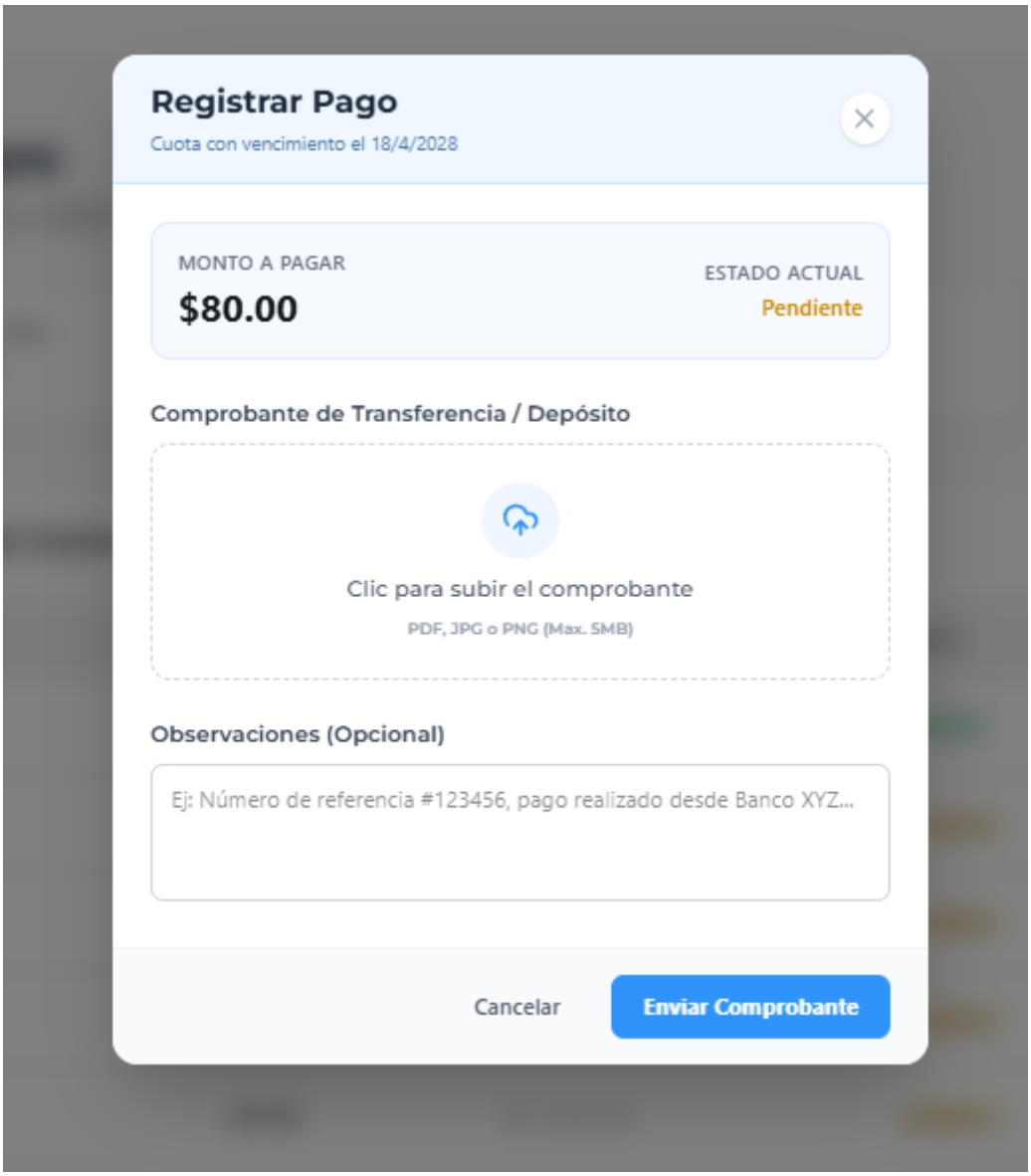
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 39. Interfaz de control de pagos



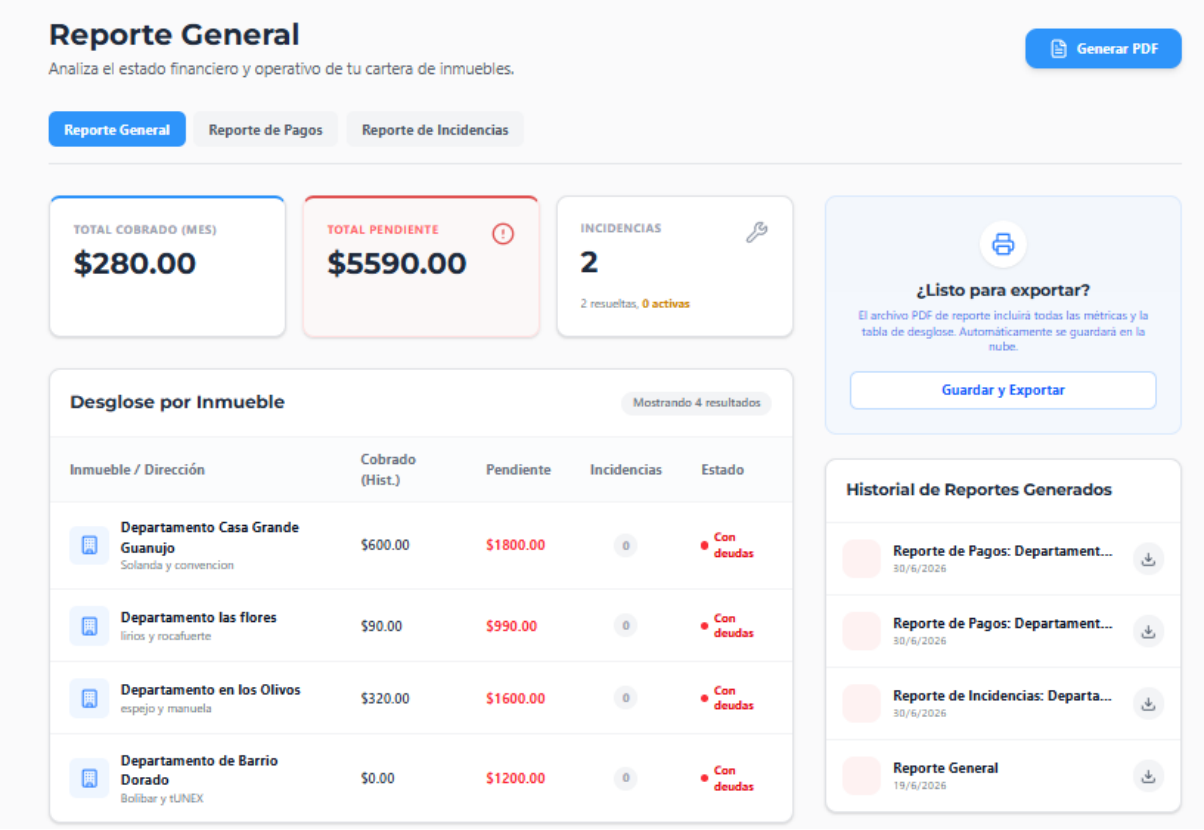
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 40. ***Interfaz de registro de pagos***



Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 41. Interfaz de generación de reportes



Nota. Elaborado por: Andrade P. Kevin (2026).

Controlador

Funciona como el cerebro que coordina las peticiones; cuando un inquilino sube un comprobante de pago, el controlador recibe esa señal, verifica la autenticidad con el servicio de seguridad y ordena al modelo que actualice el estado del alquiler.

Algoritmo de autenticación (Login) y enrutamiento de roles

Recibe las credenciales de usuario desde el frontend y verifica su existencia en la base de datos utilizando el facade Hash de laravel para la comprobación de contraseñas. Además de autenticar al usuario, extrae el parámetro `rol_tipo` para determinar y enrutar la sesión hacia el panel correspondiente (Arrendador, Arrendatario o Control Parental), garantizando la protección y el control de acceso a las rutas del sistema.

AuthController.php -> login

```
public function login(Request $request)
{
    $validator = Validator::make($request->all(), [
        'correo' => 'required|email',
        'contrasena' => 'required|string',
    ]);

    if ($validator->fails()) {
        return response()->json(['success' => false, 'errors' => $validator->errors()], 422);
    }

    $usuario = Usuario::where('correo', $request->correo)->first();

    if (!$usuario || !Hash::check($request->contrasena, $usuario->contrasena)) {
        return response()->json(['success' => false, 'message' => 'Credenciales incorrectas'], 401);
    }

    if (!$usuario->activo) {
        return response()->json(['success' => false, 'message' => 'Cuenta inactiva'], 403);
    }

    return response()->json([
        'success' => true,
        'message' => 'Login exitoso',
        'data' => clone $usuario
    ]);
}
```

Algoritmo de vinculación de múltiples partes

Procesa la creación de contratos de alquiler conectando múltiples entidades. Relaciona los identificadores del Arrendador, Arrendatario y, de forma opcional, el Control Parental. Constituye el núcleo relacional del sistema al crear un registro en la tabla de vinculaciones que funciona como puente, permitiendo el flujo de transacciones (pagos, incidencias) entre un inmueble específico y los usuarios involucrados.

VinculacionController.php -> store

```
public function store(Request $request)
{
    // Validaciones de correos y datos

    $arrendatario = Arrendatario::whereHas('usuario', function($q) use ($request) {
        $q->where('correo', $request->correo_arrendatario);
    })->first();
```

```
    if (!$arrendatario) {
        return response()->json(['success'=>false, 'message'=>'El correo del arrendatario no pertenece a un arrendatario registrado.']);
    }

    $id_control_parental = null;
    if ($request->filled('correo_control_parental')) {
        $parental = ControlParental::whereHas('usuario', function($q) use ($request) {
            $q->where('correo', $request->correo_control_parental);
        })->first();
        if($parental) $id_control_parental = $parental->id_control_parental;
    }

    $vinculacion = Vinculacion::create([
        'id_inmueble' => $request->id_inmueble,
        'id_arrendador' => $request->id_usuario_arrendador,
        'id_arrendatario' => $arrendatario->id_arrendatario,
        'id_control_parental' => $id_control_parental,
        'valor' => $request->valor,
        'garantia' => $request->garantia,
        'fecha_inicio' => $request->fecha_inicio,
        'fecha_fin' => $request->fecha_fin,
        'normas' => $request->normas,
        'contratourl' => $request->contratourl,
        'activo' => true
    ]);

    return response()->json(['success' => true, 'data' => $vinculacion]);
}
```

Algoritmo de filtrado dinámico y seguridad

Implementa un constructor de consultas (Query Builder) aplicado en los módulos de Inmuebles, Pagos e Incidencias. Analiza el tipo de identificador recibido en la petición (id_usuario_arrendador, id_usuario_arrendatario o id_usuario_control_parental) y ejecuta subconsultas para retornar exclusivamente la información autorizada para dicho perfil. Garantiza la seguridad horizontal del sistema al impedir el acceso no autorizado a registros pertenecientes a otros usuarios.

PagoController.php -> index

```
public function index(Request $request)
{
    $query = Pago::with(['vinculacion.inmueble', 'vinculacion.arrendatario.usuario']);
```

```
    if ($request->has('id_usuario_arrendador')) {
        $arrendador = Arrendador::where('id_usuario', $request->id_usuario_arrendador)->first();
        if ($arrendador) {
            $query->whereHas('vinculacion', function ($q) use ($arrendador) {
                $q->where('id_arrendador', $arrendador->id_arrendador);
            });
        } else {
            return response()->json(['success' => true, 'data' => []]);
        }
    }

    if ($request->has('id_usuario_control_parental')) {
        $parental = ControlParental::where('id_usuario', $request->id_usuario_control_parental)->first();
        if ($parental) {
            $query->whereHas('vinculacion', function ($q) use ($parental) {
                $q->where('id_control_parental', $parental->id_control_parental);
            });
        }
    }

    return response()->json(['success' => true, 'data' => $query->orderBy('fecha_vencimiento', 'desc')->get()]);
}
```

Proxy transaccional de archivos nube

Gestiona la transferencia de archivos binarios (fotografías, comprobantes, documentos PDF) enviados desde el frontend hacia los buckets de almacenamiento en Supabase. Emplea el cliente HTTP de laravel para procesar el archivo, adjuntar encabezados de autenticación y retornar la URL pública del recurso. Optimiza la base de datos PostgreSQL al asegurar que solo se almacenen cadenas de texto (URLs) en lugar de datos binarios pesados.

SupabaseStorageController.php -> upload

```
public function upload(Request $request)
{
    // Validaciones de Request
    $file = $request->file('file');
    $bucket = $request->input('bucket');
    $path = $request->input('path');

    $fileContent = file_get_contents($file->getRealPath());
    $mimeType = $file->getMimeType();

    $url = env('SUPABASE_URL') . "/storage/v1/object/$bucket/$path";
```

```
    $response = Http::withHeaders([
        'Authorization' => 'Bearer ' . env('SUPABASE_KEY'),
        'Content-Type' => $mimeType,
    ])->send('POST', $url, [
        'body' => $fileContent
    ]);

    if ($response->successful()) {
        $publicUrl = env('SUPABASE_URL') . "/storage/v1/object/public/$bucket/$path";
        return response()->json(['success' => true, 'public_url' => $publicUrl]);
    }

    return response()->json(['success' => false, 'message' => 'Error al subir el archivo'], 500);
}
```

Este enfoque modular permite que si más adelante queremos cambiar cómo se ve la interfaz, no afectamos cómo funcionan los datos internos y si algo falla en la lógica, sabemos que probablemente está en el modelo, no en los botones. El patrón MVC nos ayuda a trabajar en paralelo y a no tener un código tan enredado. Básicamente, cada quien hace lo suyo.

4.3 Fase III: Codificación

El desarrollo técnico del sistema GestInWEB se estructuró bajo una arquitectura de tres capas, lo que permite separar la interfaz de usuario, la lógica de negocio y la persistencia de datos. Esta división facilita el mantenimiento independiente de cada módulo y optimiza el rendimiento global de la aplicación.

Tabla 41. Estructura de Codificación

Capa	Tecnologías	Rol en el Sistema
Frontend	React.js, Vite, Tailwind CSS	Renderizado de las interfaces de usuario distribuidas por roles (arrendador, arrendatario y control parental) y consumo de servicios API.
Backend	PHP, Laravel	Procesamiento de la lógica de negocio, control de acceso mediante tokens, validaciones y exposición de endpoints REST.

Base de Datos	PostgreSQL (Supabase)	Almacenamiento relacional estructurado y gestión de archivos físicos mediante buckets de almacenamiento aislado.

Nota. Elaborado por: Andrade P. Kevin (2026).

Capa de presentación (Frontend)

Implementación de vistas modulares y manejo de estado local para la captura de datos de los formularios.

```
// Componente para la vinculación de un nuevo arrendamiento
import { useState } from 'react';
import { useAuth } from '../../contexts/AuthContext';

export default function VincularPage() {
  const { user } = useAuth();
  const [correoArrendatario, setCorreoArrendatario] = useState('');

  const handleVinculacion = (e) => {
    e.preventDefault();
    // Lógica para enviar datos a la API de Laravel
  };
}
```

Capa de lógica de negocio (Backend)

Controladores especializados encargados de recibir las peticiones HTTP, validar los datos de entrada y procesar las transacciones financieras.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Pago;

class PagoController extends Controller {
    public function store(Request $request) {
        $validated = $request->validate([
            'id_vinculacion' => 'required|string|max:200',
            'monto' => 'required|numeric',
            'tipo_pago' => 'required|string'
        ]);

        $pago = Pago::create($validated);
```

```
        return response()->json(['message' => 'Pago registrado correctamente', 'data' => $pago], 201);
    }
}
```

Capa de datos (PostgreSQL)

Definición de las estructuras de las tablas principales con restricciones de integridad y llaves primarias alfanuméricas.

```
TABLA: USUARIOS
Gestión de perfiles y credenciales de acceso
CREATE TABLE IF NOT EXISTS public.tabla_usuarios
(
    id_usuario integer NOT NULL DEFAULT nextval('tabla_usuarios_id_usuario_seq'::regclass),
    cedula character varying(20) COLLATE pg_catalog."default" NOT NULL,
    nombres character varying(100) COLLATE pg_catalog."default" NOT NULL,
    apellidos character varying(100) COLLATE pg_catalog."default" NOT NULL,
    correo character varying(150) COLLATE pg_catalog."default" NOT NULL,
    contrasena character varying(255) COLLATE pg_catalog."default" NOT NULL,
    rol_tipo character varying(50) COLLATE pg_catalog."default" NOT NULL,
    activo boolean DEFAULT true,
    telefono character varying(50) COLLATE pg_catalog."default",
    fecha_registro date,
    direccion character varying(255) COLLATE pg_catalog."default" NOT NULL DEFAULT 'Sin dirección'::character varying,
    fechnac date NOT NULL DEFAULT '1900-01-01'::date,
    foto_perfil text COLLATE pg_catalog."default",
    CONSTRAINT tabla_usuarios_pkey PRIMARY KEY (id_usuario),
    CONSTRAINT tabla_usuarios_cedula_key UNIQUE (cedula),
    CONSTRAINT tabla_usuarios_correo_key UNIQUE (correo)
);
```

Infraestructura y configuración de la base de datos (PostgreSQL)

La persistencia de la información relacional de la aplicación se gestiona de manera centralizada en el motor de base de datos PostgreSQL, administrado de forma directa a través de la herramienta pgAdmin. Por su parte, la plataforma Supabase se integra de manera externa y exclusiva para la persistencia de archivos binarios, documentos y archivos multimedia (mediante Supabase Storage). Esta arquitectura descentralizada asegura que los datos estructurados y los archivos físicos permanezcan aislados, optimizando el rendimiento del servidor principal.

La interacción entre la capa del backend (Laravel) y el motor de base de datos se realiza mediante el driver PDO de PHP, garantizando un canal seguro, eficiente y capaz de soportar las peticiones concurrentes de arrendadores, arrendatarios y usuarios de control parental.

Configuración de variables de entorno

Los parámetros de red y las credenciales requeridas para establecer la conexión con el entorno de PostgreSQL se definen y centralizan en el archivo de configuración global.env:

```
# Base de datos PostgreSQL
DB_CONNECTION=pgsql
DB_HOST=
DB_PORT=
DB_DATABASE=
DB_USERNAME=
DB_PASSWORD=

# Supabase
SUPABASE_URL=
SUPABASE_SERVICE_ROLE_KEY=
```

Estructura de contratos y vinculaciones

Las relaciones multipar titas entre los inmuebles, propietarios, estudiantes y los encargados de control parental se consolidan en la tabla de vinculaciones:

```
TABLA: VINCULACIONES (Contratos multipar titos)
CREATE TABLE tabla_vinculaciones (
   id_vinculacion VARCHAR(200) PRIMARY KEY,
   id_inmueble INTEGER NOT NULL REFERENCES tabla_inmueble(id_inmueble) ON DELETE CASCADE,
   id_arrendador INTEGER NOT NULL REFERENCES tabla_arrendador(id_arrendador),
   id_arrendatario INTEGER NOT NULL REFERENCES tabla_arrendatario(id_arrendatario),
   id_control_parental INTEGER REFERENCES tabla_controlparental(id_control_parental),
   valor DECIMAL(10,2) NOT NULL,
   activo BOOLEAN DEFAULT true
);
```

Organización de rutas de la API (web.php)

El ruteo de la aplicación se configuró de forma modular bajo el patrón Modelo-Vista-Controlador (MVC) en el backend. Cada bloque de endpoints se encuentra protegido y aislado según las responsabilidades requeridas por el sistema.

Tabla 42. Rutas api

Prefijo de Ruta	Controlador Encargado	Funcionalidad Principal
/api/auth	AuthController	Control de inicio de sesión, validación de credenciales y asignación de vistas según el rol del usuario.
/api/inmuebles	InmuebleController	Administración del catálogo de propiedades, registros fotográficos y actualización del estado de las habitaciones.
/api/vinculaciones	VinculacionController	Gestión y formalización de contratos entre el arrendador, arrendatario y control parental.
/api/pagos	PagoController	Procesamiento de transacciones, registro de montos y carga de archivos de comprobantes digitales.

Nota. Elaborado por: Andrade P. Kevin (2026).

Una vez consolidada la infraestructura base, el diseño del esquema relacional y el mapeo de las rutas de comunicación del sistema GestInWEB, se procedió a ejecutar la fase de control de calidad mediante el diseño y la ejecución de pruebas funcionales para asegurar la estabilidad del software.

4.4 Pruebas

Tras concluir las fases de diseño y codificación bajo la metodología Extreme Programming (XP), se procedió a ejecutar el ciclo de control de calidad con el fin de asegurar la estabilidad del software, la integridad de los datos y una óptima experiencia de usuario. Estas evaluaciones se desplegaron de manera estructurada en un entorno local controlado. Este proceso abarcó pruebas unitarias, de integración, de rendimiento y de seguridad, partiendo de la verificación de la lógica de negocio aislada hasta la validación de la resistencia de la infraestructura. A

continuación, se detallan el enfoque y la instrumentación de cada uno de los tipos de prueba implementados en el sistema GestInWEB.

Pruebas unitarias

El objetivo primordial de las pruebas unitarias fue validar de forma completamente aislada el correcto funcionamiento de los métodos y componentes individuales de la lógica de negocio, omitiendo dependencias externas. Se utilizó el framework Vitest junto con React Testing Library para validar el comportamiento de las funciones principales y más importantes y controladores lógicos. Componentes evaluados: Se probaron funciones críticas como el cifrado de contraseñas mediante el mecanismo hash, el cálculo exacto de los montos lógicos de deudas en los controladores, y las restricciones de validación en la recepción de datos de formularios. Resultado: Se garantizó que los algoritmos base respondieran con los tipos de datos y códigos de estado HTTP correctos ante escenarios tanto ideales como erróneos.

AuthContext

Es el corazón del estado de sesión del sistema. Controla si el usuario está logueado, guarda sus datos en localStorage para que la sesión persista al recargar la página, y expone funciones globales (login y logout) que el resto de la aplicación puede utilizar, las pruebas que se realizaron fueron:

1. Validar que inicia en estado "No autenticado" por defecto.
2. Validar que al hacer "login", el estado global cambia a "Autenticado" y se guardan los datos correctamente en el localStorage.
3. Validar que al hacer "logout", el estado se limpia y el localStorage queda vacío.

ProtectedRoute

Esta función envuelve cualquier página que deba ser privada como CobrosPage o InmueblesPage para evitar la filtración de datos, revisando quién es el usuario actual y qué rol tiene antes de ir a la ruta correspondiente, las pruebas realizadas fueron:

1. Intrusión: Si alguien intenta acceder sin estar logueado, verifica que es expulsado y redirigido a la pantalla de Login (/).

2. Acceso Autorizado: Si un usuario tiene el rol correcto (ej. un arrendador intentando entrar a su panel), verifica que se renderiza el contenido sin problema.
3. Roles Cruzados: Si un usuario intenta acceder a una ruta que no le pertenece (ej. un arrendatario queriendo entrar al panel de arrendador), valida que sea redirigido a su propio panel correspondiente (/arrendatario/inmueble).
4. Cambio de Contraseña Forzado: Si la cuenta marca debe_cambiar_password: true, valida que se le prohíba navegar libremente y sea redirigido a su pantalla de perfil forzosamente.

Componente RegisterPage

Se le realizó pruebas a la interfaz de registro de nuevos usuarios, la cual tiene como función renderizar un formulario interactivo que permite seleccionar el rol (arrendatario, arrendador, parental), muestra y oculta campos dinámicamente según el rol (ej. "Carrera Universitaria" para arrendatario, "Parentesco" para parental), realiza validaciones estrictas del lado del cliente (los correos deben coincidir, los términos de servicio deben estar aceptados), envía los datos a tu API en /api/auth/register, maneja los estados de carga (loading), renderiza errores provenientes del servidor (ej. "La cédula ya está registrada") y muestra una interfaz de éxito que espera 3 segundos antes de redirigirte automáticamente a la página de inicio de sesión.

Tabla 43. Pruebas Unitarias con Vitest

Caso de Prueba	Objetivo	Descripción del Procedimiento	Resultado esperado	Estado
Renderizado Inicial	Validar la carga correcta de la interfaz por defecto.	Acceder a la ruta del formulario de registro y verificar la visualización inicial de los elementos gráficos.	El formulario debe cargar con el perfil de "Arrendatario" seleccionado por defecto, mostrando visiblemente el campo obligatorio de **Carrera Universitaria**.	**Aprobado**
Interactividad Dinámica	Verificar el cambio de	Interactuar con los botones de selección de rol y hacer clic	El campo de Carrera Universitaria debe desaparecer de	**Aprobado**

	estructura según el rol.	específicamente en la opción **Parental**.	forma dinámica de la interfaz y, en su lugar, debe renderizarse el campo de **Parentesco**.	
Validación de Correos	Auditar la concordancia en la confirmación de identidad.	Introducir dos direcciones de correo electrónico distintas en los campos de "Correo Electrónico" y "Confirmar Correo" e intentar procesar el formulario.	El sistema debe bloquear el envío de la petición de forma inmediata y desplegar en la pantalla el mensaje de error de correos no coincidentes.	**Aprobado**
Validación de Términos	Forzar el control de políticas de privacidad.	Diligenciar todos los campos obligatorios del formulario e intentar enviarlo dejando el checkbox de aceptación de políticas sin marcar.	La plataforma debe detener el procesamiento y lanzar el aviso restrictivo que indica que la aceptación de los términos es obligatoria.	**Aprobado**
Flujo de Éxito (Camino Feliz)	Evaluar la ruta de registro exitosa y la navegación temporizada.	Simular (mockear) la API del backend para forzar una respuesta satisfactoria 200 OK, llenar el formulario de forma correcta y hacer clic en "Crear cuenta".	El sistema debe: 1. Cambiar la interfaz a la vista de "¡Registro Exitoso!". 2. Mantener la pantalla estática esperando pacientemente el umbral de 3 segundos. 3. Ejecutar de forma automática la redirección de la ruta mediante el método Maps('/').	**Aprobado**

Manejo de Errores del Servidor	Comprobar la resiliencia de la interfaz ante fallas del backend.	Simular un escenario donde la API rechaza la solicitud de registro (ej. detección de un número de cédula o correo electrónico duplicado).	El formulario de la capa de presentación debe atrapar la excepción enviada desde el backend, procesar el objeto de error y pintar en la pantalla los mensajes informativos en color rojo.	**Aprobado**

Nota. Elaborado por: Andrade P. Kevin (2026).

Pruebas de integración

Estas son pruebas de integración muy completas porque no solo validan que la interfaz se dibuje correctamente, sino que verifican el flujo completo de datos: cómo los componentes reaccionan a las acciones del usuario, cómo actualizan sus estados internos como listas y selectores y cómo finalmente empaquetan esos datos para enviarlos a la API correspondiente.

RegistroInmueblePage

Es el formulario mediante el cual un arrendador da de alta una nueva propiedad en el sistema. Recopila información básica como el título, dirección, servicios, etc, y permite armar una lista dinámica de inventario agregando y quitando ítems como "Camas", "Sillas", adicionalmente permite subir una fotografía, y de esta manera envía toda esta estructura compleja al backend en /api/inmuebles.

Para validar el comportamiento del módulo de administración de propiedades en la capa de presentación, se ejecutó un segundo ciclo de pruebas funcionales automatizadas en el navegador, emulando las acciones de gestión que realizaría un arrendador real:

Tabla 44. Prueba de integración a la interfaz RegistroInmueblePage

Caso de Prueba	Objetivo	Descripción del Procedimiento	Resultado esperado	Estado

Renderizado Inicial	Verificar la carga íntegra de la interfaz de usuario.	Acceder a la sección de registro de propiedades y evaluar el despliegue del diseño visual dinámico.	El layout principal de la página debe cargar de forma correcta, mostrando distribuidos y listos para interactuar todos los bloques modulares de características generales, dimensiones, servicios e inventario.	**Aprobado**
Interactividad Compleja (Inventario)	Validar la manipulación de arreglos dinámicos en memoria local.	Simular la inserción del nombre de un elemento (ej: "Silla"), definir la cantidad (ej: "4"), hacer clic en el botón "Añadir", y posteriormente hacer clic en el botón de eliminación (ícono del basurero).	El sistema debe agregar el registro de forma inmediata reflejándolo en la lista visible en pantalla y, tras presionar el botón de eliminación, debe retirarlo del arreglo de datos limpios actualizando la interfaz gráfica al instante.	**Aprobado**
Flujo Completo (Happy Path)	Confirmar el envío óptimo y la estructura	Completar todos los campos del formulario de inmueble de forma	El sistema debe: 1.Validar matemáticamente que el objeto JSON sea	**Aprobado**

	de los datos lógicos.	correcta, interceptar (mockear) la petición HTTP hacia el backend para retornar un estado exitoso 200 OK, y proceder con el envío formal de los datos.	estructurado y enviado hacia la API contenga con precisión los strings y enteros tipeados de forma manual. 2. Renderizar con éxito en el navegador el modal informativo de "¡Inmueble Registrado!".	

Nota. Elaborado por: Andrade P. Kevin (2026).

VincularPage

Esta es la vista donde el arrendador vincula a un estudiante a uno de sus inmuebles para formalizar el contrato. Cuando la página carga, primero hace una petición GET para traer los inmuebles disponibles del arrendador. Luego, el usuario llena datos sensibles como correos, montos, fechas, puede subir el PDF del contrato, y opcionalmente puede activar el "Control Parental". Al final, envía todo a /api/vinculaciones.

Tabla 45. Prueba de integración a la interfaz VincularPage

Caso de Prueba	Objetivo	Descripción del Procedimiento	Resultado esperado	Estado
Carga Asíncrona	Evaluar la correcta sincronización y consumo del hook de efecto inicial.	Interceptar la llamada asíncrona que ejecuta el método useEffect al cargar la página, inyectando un	El sistema debe manejar la espera del estado de carga de manera asíncrona y, posteriormente, renderizar e integrar	**Aprobado**

		arreglo de respuesta falsa con dos inmuebles de prueba.	con precisión ambos inmuebles dentro del menú desplegable (select) de la interfaz.	
Lógica condicional	Verificar el comportamiento dinámico de la interfaz según los requerimientos lógicos del formulario.	Evaluar el estado inicial del DOM y, posteriormente, activar el checkbox interactivo denominado "Control Parental".	El campo de entrada de texto correspondiente al "Correo del Control Parental" no debe existir en la pantalla de forma inicial, pero debe renderizarse y mostrarse automáticamente en la vista al marcar la casilla de verificación.	**Aprobado**
Envío y Validación	Comprobar la ejecución secuencial de peticiones HTTP y la consistencia en la construcción del payload de datos.	Simular el flujo transaccional completo: recuperar los inmuebles de la API, seleccionar una propiedad de la lista, ingresar un correo electrónico válido, definir montos y fechas vigentes, y proceder con el	El sistema debe coordinar de forma secuencial las peticiones hacia el servidor y validar con rigor matemático que el objeto JSON del payload contenga exactamente el monto y el correo ingresados, finalizando con el despliegue visual del	**Aprobado**

		envío del formulario mediante una petición POST.	mensaje de "¡Vinculación Exitosa!".	

Nota. Elaborado por: Andrade P. Kevin (2026).

Pruebas de mantenibilidad y seguridad

Para garantizar que el código fuente de la plataforma GestInWEB cumpla con los estándares de la industria, mitigue riesgos técnicos y reduzca la deuda técnica antes de su despliegue, se integró la herramienta de análisis estático SonarQube Community Edition. La configuración inicial del escaneo se centralizó en la raíz del proyecto mediante el archivo de propiedades sonar-project properties, definiendo los parámetros de red locales, el token de autenticación seguro y aplicando exclusiones estrictas sobre directorios de librerías externas node_modules y vendor para evitar falsos positivos y optimizar el tiempo de ejecución:

Figura 42. Archivo de propiedades de sonarQube

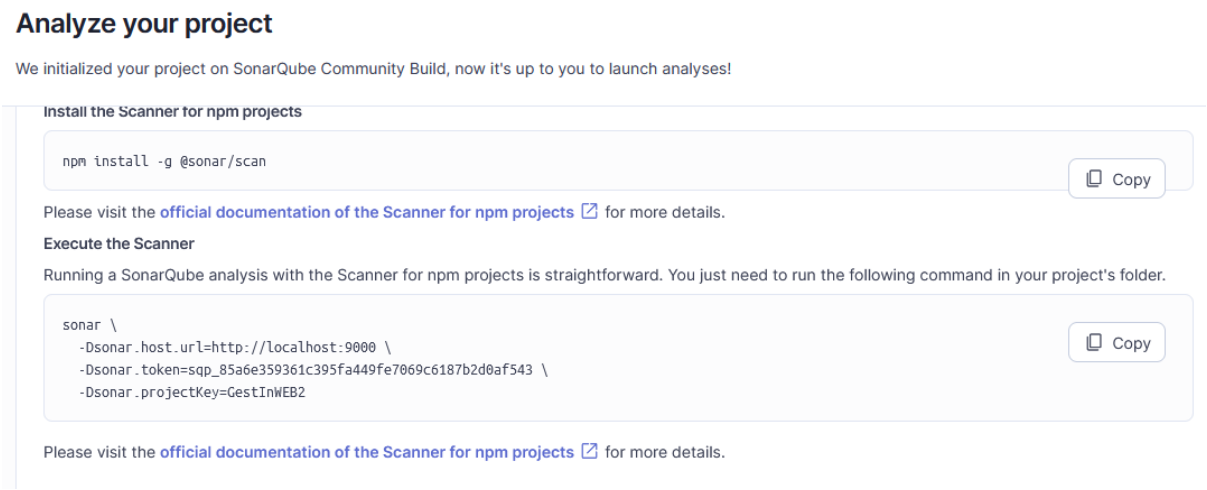
sonar-project.properties

```
1  sonar.host.url=http://localhost:9000
2  sonar.token=sqp_9cd209b05c3dd1c3b60896e7c6d33f07745bf5d9
3  sonar.projectKey=GestInWEB
4  sonar.sources=.
5  sonar.exclusions=**/node_modules/**,**/vendor/**
```

Nota. Elaborado por: Andrade P. Kevin (2026).

Se crea el proyecto en una instancia local, para que luego la herramienta nos dé una Apikey con el que inicialicemos el proyecto desde la terminal con la ruta raíz:

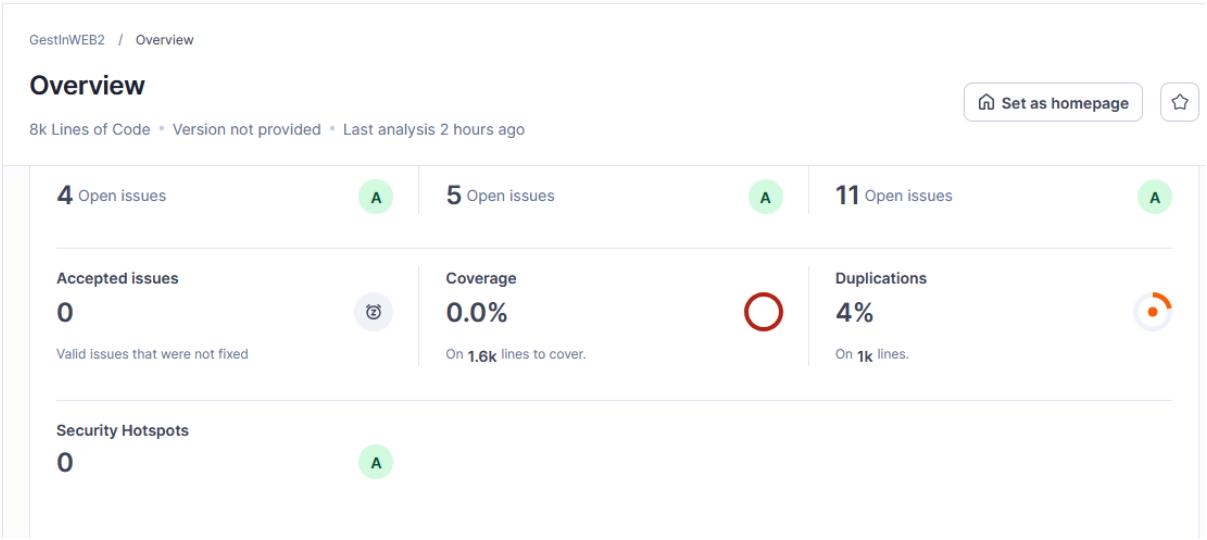
Figura 43. Comandos para inciar el escaneo



Nota. Elaborado por: Andrade P. Kevin (2026).

Al terminal el escaneo “ScannerEngine: SonarScanner Engine completed successfully” automáticamente en el dashboard de la herramienta se nos actualiza y nos da los resultados tras el análisis.

Figura 44. Vista de resultados del escaneo



Nota. Elaborado por: Andrade P. Kevin (2026).

El motor de inspección analizó un total de 8,000 líneas de código pertenecientes a la arquitectura modular del sistema, determinando un estado general APROBADO en el umbral

de calidad Quality Gate. A continuación, se detallan las métricas e indicadores arrojados en la evaluación del código global Overall Code:

Tabla 46. Métricas e indicadores arrojados del escaneo

Dimensión de Calidad	**Incidentes Detectados**	**Calificación (Rating)**	**Descripción e Impacto Técnico**	**Corregido**
Seguridad (*Security*)	4 problemas abiertos	**A**	Identifica vulnerabilidades potenciales en el flujo lógico. Requiere revisión de sanitización de entradas para blindar el sistema frente a riesgos de inyección.	SI
Confiabilidad (*Reliability*)	5 problemas abiertos	**A**	Evalúa la presencia de *bugs* lógicos menores que podrían provocar fallos en tiempo de ejecución bajo escenarios concurrentes.	SI
Mantenibilidad (*Maintainability*)	11 problemas abiertos	**A**	Mide la limpieza del código y la facilidad para recibir modificaciones futuras. La calificación **A** demuestra una baja deuda técnica en los controladores de Laravel y componentes de React.	SI
Duplicación (*Duplications*)	4% del código		Porcentaje óptimo que valida la aplicación del rigor técnico de Extreme Programming (XP), evidenciando un proceso de refactorización constante.	SI
Cobertura	0,0%		Indica las líneas de código cubiertas por pruebas unitarias automáticas. Al ejecutarse el escaneo sobre	

			archivos locales puros, el indicador se mantiene al mínimo, delegando la validación a la matriz de pruebas funcionales de interfaz.	

Nota. Elaborado por: Andrade P. Kevin (2026).

Este análisis estático ratifica que la plataforma es altamente mantenible y posee una estructura sólida. El estado Passed certifica que el software es técnicamente viable, seguro y modular, quedando listo para su implementación operativa

Análisis crítico y arquitectura de la seguridad transaccional

El pilar de seguridad del sistema GestInWEB se diseñó bajo el principio de Defensa en Profundidad, estructurando salvaguardas superpuestas tanto en la capa de presentación React.js como en el motor de ejecución del backend Laravel, el esquema de control se estructuró en cinco dimensiones de protección:

1. Autenticación robusta y gestión de información mínima

El proceso de inicio de sesión actúa como la primera barrera perimetral del sistema. Las pruebas automatizadas en esta unidad validan que el sistema rechace de forma hermética cualquier intento de acceso no autorizado mediante la aplicación de tres reglas estrictas:

Validación de Presencia: Se verifica que la petición contenga obligatoriamente el correo electrónico y la contraseña lógicos; ante la ausencia de cualquiera de las variables, el flujo se interrumpe inmediatamente.

Mensajes de Error Opacos: Si las credenciales son erróneas, el backend retorna un código de estado 401 Unauthorized acompañado de un mensaje genérico. Esta práctica mitiga los ataques de enumeración de usuarios, impidiendo que un atacante deduzca si el correo electrónico existe en la base de datos PostgreSQL.

Control de Estado de Cuenta: El sistema valida el atributo booleano activo en la tabla de usuarios. Si un perfil ha sido deshabilitado administrativamente, el controlador bloquea el acceso emitiendo un código 403 Forbidden, impidiendo que credenciales previamente válidas vulneren el entorno.

2. Cifrado criptográfico de la persistencia

Para asegurar la confidencialidad de la información de la comunidad universitaria de la Universidad Estatal de Bolívar, el sistema prohíbe de forma absoluta el almacenamiento de datos sensibles en texto plano.

Mecanismo de Hash: El backend utiliza el algoritmo de derivación de claves Bcrypt a través del facade Hash de Laravel. Este método añade un vector de salado aleatorio por cada registro, volviendo computacionalmente inviable el descifrado mediante tablas de búsqueda precomputadas llamada Rainbow Tables.

Integridad ante Fugas: Incluso ante un escenario hipotético de compromiso directo sobre el gestor pgAdmin, las contraseñas expuestas conservan su propiedad de irreversibilidad, protegiendo la identidad digital de los usuarios en el ecosistema de la parroquia Guanujo.

3. Validación estricta de identidad y unicidad en casos de uso

La integridad referencial de la base de datos se protege mediante restricciones lógicas de nivel superior directamente en el esquema de tablas de PostgreSQL:

Algoritmo de identidad oficial: Al registrar un nuevo usuario, el sistema evalúa la validez de la cédula de ciudadanía ecuatoriana aplicando un validador de longitud de 10 dígitos, códigos de provincia válidos y el cálculo algorítmico del dígito verificador utilizando el módulo 10.

Restricciones de unicidad (UNIQUE): Para mitigar la suplantación de identidad y la redundancia de datos, los campos cédula y correo se encuentran configurados con índices de restricción única en el motor relacional, si un estudiante o arrendador intenta registrar un dato preexistente, el driver PDO de la base de datos detiene la transacción, impidiendo la duplicidad del expediente.

4. Control de acceso basado en roles (RBAC)

La plataforma implementa un ruteo protegido por un middleware de seguridad en el backend, aislando las funcionalidades de manera estricta según el tipo de rol verificado:

Permisos restringidos: Un usuario con rol Arrendatario posee acceso exclusivo para consultar sus estados de cuenta e ingresar tickets de incidencias propios.

Aislamiento horizontal de datos: Mediante el algoritmo de filtrado dinámico y seguridad, el sistema evalúa activamente el identificador del usuario en cada petición HTTP. Un arrendatario o un perfil de Control Parental tienen prohibido de forma absoluta consultar, editar o manipular contratos, inmuebles o transacciones financieras de terceros, bloqueando la manipulación maliciosa de URLs o parámetros en el frontend.

5. Gestión de archivos binarios aislados

La seguridad de los documentos sensibles como contratos digitales en formato PDF o capturas de transferencias se delega a una arquitectura descentralizada:

Proxy transaccional: El frontend nunca interactúa directamente con las credenciales de almacenamiento. El backend de laravel actúa como un proxy seguro, inyectando los encabezados de autenticación Authorization Bearer originados de variables de entorno protegidas de nivel de sistema.

Sanitización de archivos: Se restringe la subida de elementos mediante validaciones de tipo MIME, impidiendo la inyección de scripts maliciosos o ejecutables encubiertos en los buckets de Supabase Storage.

Validación de capas de seguridad en la API

A continuación, se tabula el desglose técnico de las capas lógicas implementadas en el controlador de autenticación del backend (AuthController), identificando el tipo de validación realizada y el vector de ataque informático que previene de forma directa dentro de la plataforma **GestInWEB**:

Tabla 47. Validación de Capas de Seguridad en la API

Capa	Validación Lógica Implementada	Punto final / Componente	Vector de Ataque Prevenido

1	Campos obligatorios y formato email	AuthController -> login	Bypass de validación y denegación de servicio por payloads vacíos.
2	Constructor de consultas relacionales (Query Builder)	Usuario::where('correo', ...)	Inyección SQL (SQLi) por manipulación de parámetros de entrada.
3	Retorno de códigos de estado y mensajes opacos	Excepción401 Unauthorized	Enumeración de usuarios e ingeniería social.
4	Verificación criptográfica de hash (Hash::check)	Validación contra la columnacontrasena	Descifrado por fuerza bruta y ataques de diccionario.
5	Evaluación de estado booleano de la cuenta	Condicionalif (!$usuario->activo)	Acceso ilícito por perfiles suspendidos o dados de baja.
6	Seguridad horizontal mediante filtrado dinámico	PagoController -> index	Inyección de identificadores indirectos a objetos (IDOR).
7	Aislamiento de variables críticas de configuración	Archivo de entorno .env local	Exposición de credenciales de infraestructura

			(*Information Disclosure*).
8	Proxificación de archivos con cabeceras Bearer	SupabaseStorageController -> upload	Manipulación, secuestro o alteración de objetos en la nube de terceros.

Nota. Elaborado por: Andrade P. Kevin (2026).

Tras concluir las fase de pruebas bajo la metodología Extreme Programming, el ciclo de control de calidad se enfocó en asegurar la estabilidad del software, la integridad de los datos y una experiencia clara para el usuario dentro de un entorno local controlado, cubriendo pruebas unitarias con Vitest y React Testing Library que validaron funciones críticas como el cifrado y el cálculo de deudas junto a la gestión de sesiones en AuthContext, integrando pruebas de navegador que emularon acciones reales en componentes como RegistroInmueblePage y VincularPage para confirmar que el flujo de datos desde la interfaz hacia la API y la sincronización asíncrona funcionen correctamente en los escenarios de uso diario.

Para cumplir con los estándares técnicos y reducir la deuda del sistema, integramos la herramienta SonarQube que tras el análisis de 8,000 líneas de código otorgó una calificación A en el Quality Gate certificando una arquitectura lista para su despliegue operativo, mientras que la seguridad se construyó mediante el principio de defensa en profundidad combinando algoritmos Bcrypt para proteger las credenciales, reglas estrictas en PostgreSQL contra la duplicidad, control de acceso basado en roles que garantiza el aislamiento de datos entre usuarios y un proxy transaccional para gestionar archivos PDF, lo cual blinda a la plataforma GestInWEB ante posibles inyecciones o intentos de manipulación maliciosa de información.

CONCLUSIONES

- El sistema web GestInWEB cumplió con los objetivos técnicos del proyecto al centralizar y automatizar los flujos de trabajo administrativos, financieros y de gestión de incidentes. Reemplaza los procesos tradicionales del mercado de alojamiento estudiantil de la parroquia de Guanujo con un entorno digital estructurado y transparente.
- Las evaluaciones de campo realizadas con estudiantes y propietarios de la parroquia de Guanujo identificaron problemas operativos derivados de registros manuales y acuerdos verbales. Las historias de usuario, estructuradas mediante Programación Extrema, clarificaron los requisitos del software y aseguraron que el sistema abordara la informalidad y la desorganización financiera dentro de la comunidad universitaria.
- La plataforma se modeló utilizando el patrón Modelo-Vista-Controlador y se validó con las Vistas 4+1 de Kruchten para garantizar una clara separación técnica. El esquema relacional de PostgreSQL, gestionado en pgAdmin, impuso tipos de datos de clave primaria para optimizar la indexación y prevenir conflictos de datos. La interfaz de usuario priorizó la usabilidad, incorporando componentes interactivos y un esquema de seguridad para prevenir el acceso no autorizado.
- Se desarrollaron módulos operativos utilizando React.js y Vite para la capa de presentación, y PHP con laravel para la lógica de negocio. Esto permitió una vinculación eficiente de propiedades, paneles de verificación de pagos con alertas visuales y un canal de comunicación para tickets de incidencias con soporte fotográfico. El almacenamiento de archivos grandes en buckets de Supabase mantuvo la eficiencia de la base de datos local, garantizando tiempos de respuesta inferiores a 2 segundos para transacciones complejas.
- El control de calidad validó la estabilidad del software mediante pruebas unitarias, de integración, de usabilidad, de rendimiento y de seguridad. Los flujos de trabajo automatizados se ejecutaron en menos de 2 segundos. El análisis estático del código en SonarQube confirmó la mantenibilidad, con 8000 líneas de código que superaron la prueba de calidad con una calificación "A" y una baja deuda técnica.

RECOMENDACIONES

- Monitorear continuamente las métricas del servidor, incluyendo CPU, RAM y tasas de transferencia, para optimizar la infraestructura en la nube y anticipar las necesidades de escalado de recursos durante los picos de tráfico estacionales al inicio de cada semestre académico.
- Organizar talleres prácticos con los arrendadores de inmuebles en la parroquia de Guanujo para facilitar la transición de los registros en papel a la plataforma digital y reducir la resistencia al cambio tecnológico.
- Programar auditorías periódicas dentro del gestor pgAdmin para verificar el rendimiento de la indexación de las llaves primarias y evaluar la vigencia de las directrices de seguridad horizontal, asegurando que el aislamiento de datos por perfiles permanezca hermético ante futuras modificaciones.
- Programar auditorías periódicas en pgAdmin para verificar la indexación de la clave primaria y garantizar que las directrices de seguridad horizontal mantengan un aislamiento seguro de los datos. Integrar api de mensajería, como whatsApp o sms, en el backend de laravel para enviar automáticamente alertas de pago atrasado y actualizaciones de incidencias a los estudiantes y sus tutores legales.
- Configurar un flujo de trabajo de integración y despliegue automatizado en el repositorio de código fuente, que requiera que cada actualización supere pruebas funcionales locales y umbrales de escaneo, y utilizar herramientas robustas para mantener la seguridad y la calidad del código.

BIBLIOGRAFÍA

Asamblea Nacional del Ecuador. (2016). *Código Orgánico de Economía Social de los Conocimientos, Creatividad e Innovación - INGENIOS* (Registro Oficial Suplemento 899, 9 de diciembre de 2016). https://www.registroficial.gob.ec

Asamblea Nacional del Ecuador. (2021). *Ley Orgánica de Protección de Datos Personales* (Registro Oficial Suplemento 459, 26 de mayo de 2021). https://www.registroficial.gob.ec

Asamblea Nacional del Ecuador. (2022). *Código Civil* (Última reforma: Edición Constitucional del Registro Oficial No. 15). Registro Oficial. http://biblioteca.defensoria.gob.ec/handle/37000/3410

Avilés, S., Avila-Pesantez, D., & Avila, L. (2020). Desarrollo de sistema web basado en los frameworks de Laravel y Vue.js, para la gestión por procesos: Un estudio de caso. *Revista Peruana de Computación y Sistemas, 3*(2), 3–10. https://doi.org/10.15381/rpcs.v3i2.19256

Amazon Web Services. (2022, agosto 8). *¿Qué es una interfaz de programación de aplicaciones (API)?* https://aws.amazon.com/es/what-is/api/

Ayezabu, A. Z. (2022). Supabase vs Firebase: Evaluation of performance and development of Progressive Web Apps. https://urn.fi/URN:NBN:fi:amk-2022090920032

Barahona Blanco, C. (2020). *Diseño y desarrollo de una aplicación web para la gestión de inmuebles en alquiler y el control de pagos* [Trabajo de fin de grado, Universidad Politécnica de Madrid]. https://oa.upm.es/63315/

Baumann, H. (2022). *Conoce qué es CRUD y por qué es fundamental para desarrollar sitios y aplicaciones*. Crehana. https://www.crehana.com/blog/transformacion-digital/que-es-crud/

Bustamante, D., & Rodríguez, J. (2014). *Metodología actual: Metodología XP* [Trabajo académico, Semestre VII]. Universidad Nacional Experimental de los Llanos Occidentales Ezequiel Zamora.

Cabanellas, G., & Cabanellas, G. (1979). *Diccionario jurídico elemental*. Heliasta.

Carreño Dueñas, D. A., Amaya González, L. F., Ruiz Orjuela, E. T., & Tiboche, F. J. (2019). Diseño de un sistema para la gestión de inventarios de las pymes en el sector alimentario. *Industrial Data*, *22*(1), 113–122.

Congreso Nacional del Ecuador. (2000). *Ley de Inquilinato* (Codificación 1, Registro Oficial 196, 1 de noviembre de 2000, última modificación 30 de diciembre de 2016). https://www.registroficial.gob.ec

Fernández, H. F., & Rodríguez, J. H. (2021). *Aplicaciones web con PHP*. Ediciones de la U.

Galhego, M. (2022). *HTTP: Lo mínimo que todo desarrollador web debe saber*. Babelcube Inc.

García-Ávila, S. (2017). Alfabetización digital. *Razón y Palabra*, *21*(3_98), 66–81.

Hamidli, N. (2023). Introduction to UI/UX design: Key concepts and principles. *Research Article*, *18*.

Hilliard, R. (2000). *IEEE-Std-1471-2000 recommended practice for architectural description of software-intensive systems*. IEEE. http://standards.ieee.org

Ibáñez, L. H. (2016). *Administración de sistemas gestores de bases de datos*. Ra-Ma Editorial.

Institute of Electrical and Electronics Engineers. (1990). *IEEE standard glossary of software engineering terminology*. http://dis.unal.edu.co/~icasta/GGP/_Ver_2012_1/Documentos/Normas/610-12-1990.pdf

Kruchten, P. (1995). Planos Arquitectónicos: El Modelo de 4+ 1 Vistas de la Arquitectura del Software. IEEE software, 12(6), 42-50.

Letelier, P., & Penadés, M. C. (2006). *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Universidad Politécnica de Valencia.

Lockhart, T. (1996). *Manual del usuario de PostgreSQL*. http://palomo.usach.cl/Docs/postgres/Postgres-Tutorial.pdf

Lovaton Navarro, G. H. (2019). *Implementación de un sistema web para mejorar la gestión administrativa en la atención a los estudiantes de la Asociación Educativa Hermenéutica &*

Valores – Tarma – 2018 [Tesis de grado, Universidad Nacional Daniel Alcides Carrión]. http://repositorio.undac.edu.pe/handle/undac/1742

Martínez, J. (2018, septiembre). *Sistema para la gestión de los inmuebles*. Conferencia presentada en el VIII Congreso Iberoamericano de Ingeniería de Proyectos / III Conferencia Científica Internacional UCIENCIA 2018, Universidad de las Ciencias Informáticas, La Habana, Cuba. https://repositorio.uci.cu/jspui/handle/123456789/9548

Pacheco Astudillo, C. S., & Sangoquiza Iza, E. K. (2024). *Desarrollo de una aplicación web y móvil para la gestión de inmuebles en el conjunto habitacional Los Arupos de Latacunga* [Tesis de pregrado, Universidad Técnica de Cotopaxi]. http://repositorio.utc.edu.ec/handle/27000/12030

Reinoso Alvarado, W. P., & Revelo Herrera, K. M. (2022). *Implementación de un sistema de gestión de inmuebles: Barrio Granda Centeno* [Tesis de grado, Escuela Politécnica Nacional]. https://bibdigital.epn.edu.ec/handle/15000/22880

Saldaña, R. (2010). *Ingeniería de software: Un enfoque práctico*. McGraw-Hill.

Servicio de Rentas Internas. (2004). *Ley de Régimen Tributario Interno* (última reforma 2024). https://www.sri.gob.ec

Sommerville, I. (2005). *Ingeniería del software* (7a ed.). Pearson Educación.

Toro Lazo, A., & Peláez Valencia, L. E. (2016). Ingeniería de requisitos: De la especificación de requisitos de software al aseguramiento de la calidad. *Entre Ciencia e Ingeniería*, *10*(20), 117–123. https://doi.org/10.31908/19098367.393

Trigás Gallego, M. (2012). *Metodología Scrum*. Universitat Oberta de Catalunya. https://hdl.handle.net/10609/17885

Valdettaro, S. (2007). Notas sobre la "diferencia": Aproximaciones a la "interfaz". *La Trama de la Comunicación*, *12*, 209–223.

Vilca Ayquipa, D. B., & Pulache Chanta, J. A. (2023). *Desarrollo de un sistema web para la gestión administrativa en un gimnasio, Lima 2022* [Tesis de grado, Universidad Wiener]. https://repositorio.uwiener.edu.pe/bitstream/handle/20.500.13053/8791/T061_74738388.pdf

Villalobos, G. M., Sánchez, G. D. C., & Gutiérrez, D. A. B. (2010). Diseño de framework web para el desarrollo dinámico de aplicaciones. *Scientia et Technica*, *16*(44), 178–183.

Zurita Lara, B. N., & Recalde Araujo, H. M. (2020). *Sistema web para la gestión académica y administrativa de empresa de capacitación profesional DIENAV* [Tesis de grado, Universidad Tecnológica Israel]. http://repositorio.uisrael.edu.ec/handle/47000/2489

ANEXOS

Anexo 1

Cronograma de Gantt

Figura 45. Cronograma (Gantt)



Nota. Elaborado por: Andrade P. Kevin (2026).

Anexo 2

Presupuesto Ejecutado

Tabla 48. Presupuesto Ejecutado

Categoría	Recurso / Descripción	Cantidad (Meses/Unidades)	Costo Unitario (USD)	Costo Total (USD)
Recursos Humanos	Personal de Desarrollo (Costo asignado a COCOMO)	1	0.00	0.00
Recursos de Software	Licencia de Dominio y Hosting (Anual)	1	150.00	150.00
	VS Code	1	0.00	0.00
	Visily	1	0.00	0.00
	GitHub Actions	1	0.00	0.00
Servicios y Logística	Internet	3.5	35.00	122.50
	Consumo de Energía Eléctrica	3.5	22.50	78.75
	Materiales y Útiles de Oficina	1	30.00	30.00
	Movilización y Pasajes	1	15.00	15.00
Equipos	Computadora	1	1000.00	1000.00
TOTAL, OPERATIVO				**1396.25**

Nota. Elaborado por: Andrade P. Kevin (2026).

Anexo 3

Instrumentos de recopilación de datos

ENCUESTA 1: ARRENDATARIOS (ESTUDIANTES)

Objetivo: Identificar las dificultades que tienen los estudiantes al gestionar, pagar y reportar problemas en sus cuartos o departamentos.

Sección 1: Perfil y Gestión Administrativa

1. **¿Cuánto tiempo lleva arrendando en su residencia actual?**

- Menos de 6 meses
- 6–12 meses
- 1–2 años
- Más de 2 años

2. **Seleccione: Actualmente, ¿cómo gestiona el contrato y las reglas de su alojamiento?**

- Contrato físico en papel
- Acuerdo verbal únicamente
- No tengo claro cuáles son las reglas ni el tiempo de vigencia

3. **Seleccione: ¿Qué información le gustaría tener siempre disponible de forma digital?**

- Copia digital del contrato
- Historial de mis pagos realizados
- Reglamento interno del inmueble
- Estado de mis reportes de mantenimiento

Sección 2: Gestión Financiera

4. **Seleccione: ¿Cómo realiza comúnmente el pago de su mensualidad?**

- Efectivo directamente al dueño
- Transferencia bancaria o depósito

5. **Seleccione: ¿Recibe algún comprobante de pago formal (físico o digital) cada mes?**

- Sí, siempre
- A veces
- Nunca, solo confío en que el dueño lo anote

6. **Seleccione: ¿Qué función del sistema web le resultaría más útil para gestionar sus pagos?**

- Subir una foto/captura del comprobante de transferencia
- Recibir una notificación o recibo en PDF automático cuando el dueño apruebe el pago
- Ver un estado de cuenta de deudas pendientes

7. **Seleccione: ¿Le gustaría recibir un recordatorio automático antes de la fecha límite de pago?**

- Sí
- No

- Indiferente

Sección 3: Gestión de Incidencias

8. **Seleccione: ¿Cómo reporta usted actualmente los daños en el inmueble (fugas, luz, internet)?**
- Mensaje de WhatsApp o llamada al dueño
- Le digo en persona cuando lo veo
- Lo arreglo yo mismo para evitar problemas

9. **Seleccione: ¿Cuánto tiempo suele tardar el dueño en solucionar un problema reportado?**
- El mismo día
- De 1 a 3 días
- Más de una semana

10. **Seleccione: ¿Le resultaría útil un módulo para crear "tickets" de incidencias con fotos y seguimiento?**
- Sí, mucho
- Tal vez
- No lo usaría

11. **Seleccione: En una escala del 1 al 5, ¿qué tan importante es para usted ver el estado de su reporte (pendiente, en proceso, resuelto)?**
- Escala de 1 a 5

ENCUESTA 2: ARRENDADORES (PROPIETARIOS)

Objetivo: Analizar cómo los propietarios gestionan inquilinos, pagos y mantenimiento para automatizar estos procesos en el sistema.

Sección 1: Gestión Administrativa

1. **Seleccione: ¿Cuántos inquilinos (estudiantes) administra en promedio?**

- 1 a 5
- 6 a 15
- Más de 15

2. **Seleccione: ¿Cómo lleva actualmente el registro de sus inquilinos (nombres, cédulas, contactos)?**

- En un cuaderno o agenda física
- En un archivo de Excel
- En el celular (solo contactos)

3. **Seleccione: Al registrar un nuevo estudiante, ¿qué documento considera indispensable almacenar?**

- Copia de cédula/papeleta de votación
- Matrícula de la universidad
- Datos de un garante
- Todas

Sección 2: Gestión Financiera

4. **Seleccione: ¿Qué tipo de reportes le gustaría que el sistema genere automáticamente?** *(Seleccione las más importantes)*

- Reporte mensual de ingresos totales
- Lista de estudiantes con pagos atrasados
- Historial de pagos por cada estudiante (exportable a PDF)

Sección 3: Gestión de Incidencias y Notificaciones

5. **Seleccione: ¿Bajo qué criterio prefiere usted organizar los problemas de mantenimiento reportados?**

- Por orden de llegada
- Por nivel de urgencia
- Por tipo de daño

6. **Seleccione: ¿Le gustaría que el sistema envíe notificaciones automáticas de recordatorio de pago?**

- Sí, me ahorraría tiempo
- No, prefiero personalmente

7. **Seleccione: ¿Estaría dispuesto a actualizar el estado de las reparaciones (ej: "En Proceso" o "Solucionado")?**

- Sí
- No

Preguntas y Respuestas de la Entrevista

Pregunta 1: ¿Cómo lleva el control de la información de alquiler (la lista de habitaciones disponibles o la búsqueda de vivienda)?

- **José Caisaguano (Arrendador):** Todo lo que hago lo anoto en un cuadernito. Sé quién ocupa cada habitación y cuánto deben, pero a veces pierdo las páginas o me olvido de escribir las cosas.
- **Luis Llumiquinga (Arrendador):** Creé una hoja de cálculo en Excel, pero es un dolor de cabeza volver a revisar los datos una y otra vez y referirme a las hojas cada vez que alguien llega o se va.
- **Elena Chiluisa (Arrendador):** Recuerdo casi todo de memoria y guardo los recibos en una carpeta vieja.
- **María Chimbo (Arrendador):** Todo a través de WhatsApp. Les escribo allí, pero cuando quiero encontrar algo de lo que hablamos hace mucho tiempo, es un lío encontrarlo.
- **Ana Tipán (Arrendatario):** Solo pagué el primer mes y el dueño solo me dio la llave y eso es todo, sin papeles.
- **Ramiro Iza (Arrendatario):** Ni siquiera recuerdo cuándo tengo que pagar en qué mes o en qué fecha tengo que pagar.
- **Cinthya Zambrano (Arrendatario):** Siempre quiero saber cuál es el contrato, pero no sé dónde está y el dueño no está y no puedo pedirlo.
- **Ricardo Holguín (Arrendatario):** Si el dueño arreglara los problemas con el apartamento, cuando le escribo al dueño no responde.

Pregunta 2: ¿Cómo se siente registrar a un nuevo inquilino o firmar un contrato de arrendamiento?

- **José Caisaguano (Arrendador):** Todos necesitamos actuar y cuidar los datos del estudiante, todos estamos sosteniendo y revisando y anotando que todo está en orden

en la habitación, así que a veces solo decimos algo para resolver la situación, luego inventan historias.

- **Luis Llumiquinga (Arrendador):** No tengo los datos de mis inquilinos, no tengo ningún papel. A veces olvido el apellido o el ID.
- **Elena Chiluisa (Arrendador):** Voy a la papelería y consigo estos contratos prehechos, y pierdo tiempo llenándolos a mano.
- **María Chimbo (Arrendador):** Me gustaría tener los datos de la persona debidamente almacenados en caso de emergencia o accidente.
- **Ana Tipán (Arrendatario):** Nunca obtuve ningún contrato. O si me dan un papelito, siempre termino perdiéndolo.
- **Ramiro Iza (Arrendatario):** Me siento incómodo alquilando solo verbalmente. Es mejor tener esto en papel con los datos de ambas partes.
- **Cinthya Zambrano (Arrendatario):** Me gustaría revisar mi contrato todo el día y saber en qué realmente acordamos.
- **Ricardo Holguín (Arrendatario):** Siempre hay una discrepancia. Me mudé el 5 o el 10 o cuánto dejé como depósito porque no hay forma de verificar.

Pregunta 3: ¿Cómo organizas actualmente la cobranza de los pagos mensuales y la emisión de los respectivos recibos?

- **José Caisaguano (Arrendador):** Compro esos talonarios de recibos en la tienda. Lleno el recibo a mano y le doy el comprobante al inquilino y guardo la copia al carbón.
- **Luis Llumiquinga (Arrendador):** En unos meses me transfieren y tengo que seguir revisando la cuenta bancaria para ver quién depositó y anotarlo en mi Excel. Es agotador.
- **Elena Chiluisa (Arrendador):** Me pagan en efectivo. El problema es que piden un recibo y no tengo el talonario de recibos.

- **María Chimbo (Arrendador):** Ni siquiera doy recibos. Solo les cobro y lo anoto para ellos y escribo pagado en mi pequeña agenda.

- **Ana Tipán (Arrendatario):** Quiero que me den un recibo aunque sea en el teléfono, para tener prueba de que pagué y no me cobren dos veces.

- **Ramiro Iza (Arrendatario):** A veces el dueño olvida que pagué y me cobra de nuevo. Si tuviéramos un registro que ambos pudiéramos ver sería mejor.

- **Cinthya Zambrano (Arrendatario):** Envío el dinero pero tengo que enviar la captura de pantalla por WhatsApp y seguir diciendo al dueño que ya pagué.

- **Ricardo Holguín (Arrendatario):** Mirar hacia atrás todo lo que he pagado durante el año sería una buena manera de organizar mi dinero.

Pregunta 4: ¿Cuál es tu mayor problema cuando no recuerdas la fecha de pago?

- **José Caisaguano (Arrendador):** Me da vergüenza andar cobrando. Se pasan de la fecha y me olvido de cobrarles a tiempo.

- **Luis Llumiquinga (Arrendador):** Tengo que abrir Excel y ver quién tiene que pagar esta semana.

- **Elena Chiluisa (Arrendador):** Tengo personas que están atrasadas meses. Porque no anoto bien las cosas; a veces incluso me confundo sobre cuánto dinero me deben.

- **María Chimbo (Arrendador):** Sería genial si algo les dijera que es hora de pagar para que no tenga que perseguirlos.

- **Ana Tipán (Arrendatario):** Pierdo el alquiler porque estoy tan ocupada en la universidad. Ojalá me hubieran recordado el día anterior.

- **Ramiro Iza (Arrendatario):** Me han multado por tres días porque olvido la fecha.

- **Cinthya Zambrano (Arrendatario):** Me gustaría recibir un mensaje de texto diciendo que tienes que pagar en tres días, para poder poner algo de dinero.

- **Ricardo Holguín (Arrendatario):** Me enojo cuando el dueño golpea mi puerta e intenta sacarme dinero. Me gustaría ser notificado de una manera más amable o que llegara un aviso automáticamente.

Pregunta 5: ¿Cómo se gestionan los informes de daños o solicitudes de mantenimiento en la propiedad?

- **José Caisaguano (Arrendador):** Me llaman a cualquier hora, a veces tarde en la noche porque el baño está atascado o se rompió una tubería.
- **Luis Llumiquinga (Arrendador):** Me envían mensajes por WhatsApp, pero con todos los mensajes que llegan, me olvido de llamar a un técnico y no les arreglo nada.
- **Elena Chiluisa (Arrendador):** No sé cuánto estoy gastando en arreglar cosas. Siento que el dinero se va allí y no llevo un control.
- **María Chimbo (Arrendador):** Hay inquilinos que me meten en problemas por daños que han hecho, o cosas que ya estaban mal.
- **Ana Tipán (Arrendatario):** El grifo de la ducha se rompió hace un mes y le dije al dueño y aún no lo han traído.
- **Ramiro Iza (Arrendatario):** Una foto del daño sería buena para enviar al dueño inmediatamente y mantenerla como prueba.
- **Cinthya Zambrano (Arrendatario):** Te quejas y no sabes si el dueño leerá el mensaje o enviará a alguien.
- **Ricardo Holguín (Arrendatario):** Necesitas un lugar para escribir 'esto está roto' y ver si están respondiendo a mi solicitud.

Pregunta 6: ¿Sería útil tener un resumen financiero mensual de ingresos y estado de ocupación? ***(Pregunta exclusiva para Arrendadores)***

- **José Caisaguano:** Nunca sé si realmente estoy ganando dinero, a veces no me pagan a tiempo y se olvidan.

- **Luis Llumiquinga:** Me gustaría saber cuánto dinero entró al final del año en un pequeño dibujo o algo, cuántas habitaciones estaban vacantes.

- **Elena Chiluisa:** Un resumen mensual sería mucho más fácil de entregar al contador y no confundirse.

- **María Chimbo:** Solo necesito saber quién me debe, quién está al día sin tener que ir a una calculadora y calcular las cosas.

Pregunta 7: Como inquilino, ¿crees que es necesario tener acceso exclusivo a la plataforma? ¿Por qué? ***(Pregunta exclusiva para Arrendatarios)***

- **Ana Tipán:** Quiero ver mis recibos antiguos y mi contrato sin tener que ir y hablar con el dueño todo el tiempo.

- **Ramiro Iza:** Quiero mi propia contraseña y solo el dueño y yo podemos ver lo que pago para entrar, por seguridad.

- **Cinthya Zambrano:** Quiero iniciar sesión, ver cuánto tengo que pagar este mes y saber si la puerta está rota.

- **Ricardo Holguín:** Sí, me da más tranquilidad saber que mis datos y pagos están en un sistema seguro y no solo en papel.

Pregunta 8: ¿A qué dispositivo accederías a la aplicación y qué nivel de facilidad de uso solicitarías?

- **José Caisaguano (Arrendador):** Solo uso mi celular. Ni siquiera enciendo la computadora de casa. Tendría que ser realmente fácil porque no soy bueno con la tecnología.

- **Luis Llumiquinga (Arrendador):** Uso la computadora de casa pero cuando estoy fuera me gustaría revisar rápidamente en el teléfono.

- **Elena Chiluisa (Arrendador):** Mi celular. Pero por favor, que tenga letras grandes y botones claros para no perderme.

- **María Chimbo (Arrendador):** Uso el teléfono y la computadora, pero no debería ser un problema estar usándolo. Con solo tres toques, el pago debería estar anotado y hecho.
- **Ana Tipán (Arrendatario):** 100 por ciento desde el celular.
- **Ramiro Iza (Arrendatario):** A veces en la computadora, pero principalmente en el teléfono.
- **Cinthya Zambrano (Arrendatario):** Desde el celular, pero la página debería verse bien y ajustarse bien a la pantalla, para no tener que hacer zoom para leer.
- **Ricardo Holguín (Arrendatario):** Todo en el teléfono, es lo que siempre tienes a mano.

Pregunta 9: ¿Qué tan importante es la seguridad y privacidad de la información almacenada?

- **José Caisaguano (Arrendador):** Nadie más debería saber cuánto estoy cobrando; eso debería ser entre el inquilino de esa habitación y yo.
- **Luis Llumiquinga (Arrendador):** Mis cosas no deberían borrarse. Si mi computadora se daña o pierdo mi teléfono móvil, quiero tener acceso a todo desde otro lugar y que todo siga ahí.
- **Elena Chiluisa (Arrendador):** Necesito una buena contraseña para iniciar sesión para que nadie más pueda ver mis cuentas.
- **María Chimbo (Arrendador):** Los números de mi cuenta bancaria no deberían ser visibles y deben permanecer muy privados.
- **Ana Tipán (Arrendatario):** El propietario no debería publicar mi número de teléfono (solo él debería tenerlo).
- **Ramiro Iza (Arrendatario):** Si no recuerdo la contraseña, debería ser fácil recuperarla sin mucha dificultad.
- **Cinthya Zambrano (Arrendatario):** Como ya hay un recibo de pago recibido del propietario, el propietario no puede cambiar nada al respecto.

- **Ricardo Holguín (Arrendatario):** Debemos mantener nuestra información segura. Eso es lo más importante con tantas estafas hoy en día.

Pregunta 10: ¿Cuáles son algunas características que ayudarán a optimizar la gestión diaria?

- **José Caisaguano (Arrendador):** Deberías poder descargar recibos.
- **Luis Llumiquinga (Arrendador):** Debería enviar un WhatsApp o un correo electrónico al inquilino tan pronto como note que ha pagado.
- **Elena Chiluisa (Arrendador):** Seré notificada un par de días antes de que el contrato de alguien vaya a terminar.
- **María Chimbo (Arrendador):** Debería poder separar las cosas por casas. Alquilo habitaciones en mi casa y tengo otra casa en otro lugar y quiero gestionar todo junto.
- **Ana Tipán (Arrendatario):** Debería recibir el recibo en mi correo electrónico y no tener que pedirlo.
- **Ramiro Iza (Arrendatario):** Las reglas de la casa deberían estar publicadas ahí mismo, como a qué hora se cierra la puerta, para que no haya quejas.
- **Cinthya Zambrano (Arrendatario):** Necesito que me muestre los detalles de pago que realizo.
- **Ricardo Holguín (Arrendatario):** Todo debería estar bien registrado para siempre, así que si hay una disputa algún día, puedo mostrar lo que realmente se pagó y lo que no.

Conclusión de la entrevista aplicada

Como se indica en las declaraciones, algunos de los principales problemas del modelo de gestión tradicional se identifican y discuten como problemas operativos y de pérdida de datos desde el comienzo de la entrevista. El uso de registros físicos o cuadernos en todos los dominios, la memoria y difusión de herramientas de mensajería como WhatsApp en el proceso de recolección conducen a la pérdida repetida de recolección de datos, actividades duplicadas

e incapacidad total para auditar pagos, incidencias, etc., por lo tanto, el fracaso para establecer el contrato. La ausencia de formalización es una causa importante de esto, y los acuerdos verbales o formularios preimpresos llenados a mano no proporcionan al inquilino la información que necesita de manera oportuna, lo que puede llevar a confusión sobre la fecha de entrada, el monto del depósito y la validez de los arrendamientos.

La desorganización documental afecta inmediatamente la economía de ambas partes debido a la falta de trazabilidad transaccional entre ellas, ya que el flujo de cobro no tiene un canal de validación; los propietarios se cansan de conciliar transferencias bancarias, los inquilinos no reciben recibos de pago inmediatos, la asimetría de información y el riesgo latente de cargos duplicados debido al olvido del propietario, y sin un canal estructurado, todos los inconvenientes se trasladan a la vida diaria a través de la comunicación constante y la fricción de mantenimiento. El problema de los incidentes o daños a la propiedad se maneja actualmente de manera reactiva e intrusiva a través de llamadas fuera de horario o mensajes de WhatsApp que terminan siendo ignorados, y no hay un flujo claro para reportar reparaciones o evidencia fotográfica del daño antes o después de la intervención técnica.

En conclusión, las entrevistas han demostrado que los problemas en el entorno de estudio no provienen de la disposición de los sujetos para trabajar, sino de la ausencia de un ecosistema digital integrado. Así que la transición de un método manual y disperso a una plataforma estructurada con GESTINWEB aborda claramente los desafíos en el entorno operativo y transformará las quejas de los usuarios en procesos automatizados, transparentes y seguros que se pueden utilizar desde cualquier dispositivo con conexión a internet.

Anexo 4

Foto de reuniones

Figura 46. Revisión de documento y requisitos del sistema - Ing. Edgar Ribadeneira



Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 47. Revisión de documento y requisitos del sistema – Ing Darwin Carrión



Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 48. Revisión del sistema y sus funcionalidades – Ing. Darwin Carrión



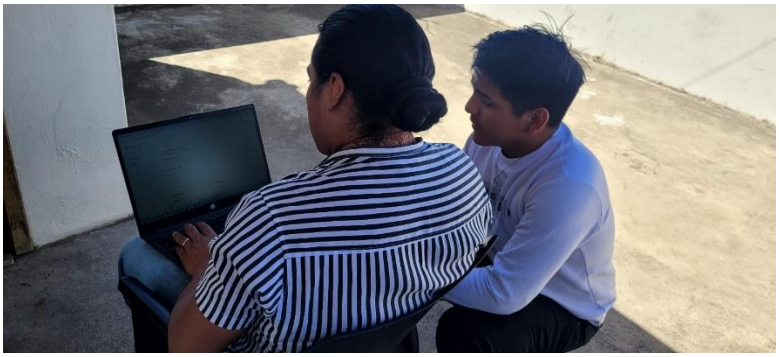
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 49. Revisión del sistema y sus funcionalidades – Ing. Edgar Ribadeneira



Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 50. Revisión del sistema y sus funcionalidades 1 - Arrendador



Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 51. ***Revisión del sistema y sus funcionalidades 2 - Arrendador***



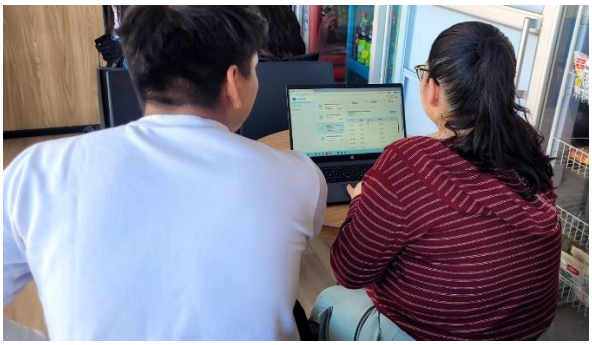
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 52. ***Revisión del sistema y sus funcionalidades 3 - Arrendador***



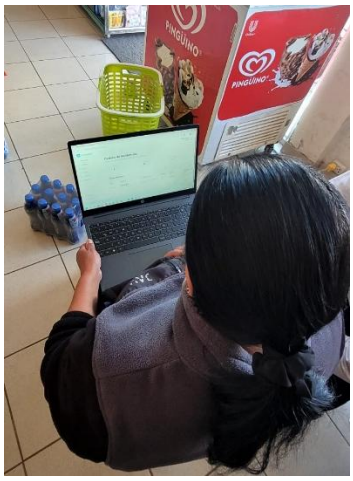
Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 53. ***Revisión del sistema y sus funcionalidades 4 - Arrendador***



Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 54. ***Revisión del sistema y sus funcionalidades 5 - Arrendador***



Nota. Elaborado por: Andrade P. Kevin (2026).

Figura 55. ***Revisión del sistema y sus funcionalidades 6 - Arrendador***



Nota. Elaborado por: Andrade P. Kevin (2026).

Anexo 5

Manual de usuario



MANUAL DE USUARIO PARA EL SISTEMA WEB INTEGRAL PARA LA GESTIÓN ADMINISTRATIVA, FINANCIERA Y DE INCIDENCIAS EN INMUEBLES DE ALQUILER PARA ESTUDIANTES DE LA UNIVERSIDAD ESTATAL DE BOLÍVAR "GestInWEB"

Autor:

KEVIN ANDRES ANDRADE PACHERRIS

Guaranda, 02 de Julio de 2026

INDICE

INTRODUCCIÓN 169

1.1 Objetivo 169

1.2 Alcance 169

1.3 Restricciones 170

INGRESO AL SISTEMA 171

Login y página principal 171

Recuperar contraseña 171

Registro de Usuario 172

Soporte, Términos, y políticas de privacidad 173

Cambio de contraseña Obligatoria 175

PROCESOS MODULO ARRENDATARIOS 177

Interfaz de la pantalla inicial del arrendatario 177

Interfaz del visualización y registro de Pagos 178

Reportes PDF 180

Interfaz para reportar incidencia 180

PROCESOS MODULO ARRENDADORES 181

Interfaz Principal de inmuebles disponibles 181

Interfaz de registro de Inmueble 182

Interfaz de vinculación de partes involucradas en el contrato 183

Interfaz de gestión de cobros 184

Validación de Pago 185

Interfaz de gestión de Incidencias.....186

Interfaz reporte General 187

PROCESOS MODULO USUARIO CONTROL PARENTAL 187

Interfaz principal Usuario de control parental 187

INDICE DE FIGURAS

Figura 1. Interfaz inicial del sistema 171

Figura 2. Interfaz recuperar contraseña 172

Figura 3. Interfaz Registro 173

Figura 4. Pie de pagina de la plataforma 173

Figura 5. Ventana de soporte técnico 174

Figura 6. Ventana de Términos y condiciones 174

Figura 7. Ventana de Políticas de seguridad 175

Figura 8. Ventana de correo enviado con exito 175

Figura 9. Ingreso al sistema con la contraseña temporal 176

Figura 10. Interfaz de configuración de Perfil 176

Figura 11. Interfaz de configuración de Perfil “Fotografía editada” 177

Figura 12. Interfaz Principal del arrendatario “vizualización inmueble” 177

Figura 13. Interfaz Principal del arrendatario “Visualización contrato” 178

Figura 14. Interfaz del visualización y registro de Pagos 178

Figura 15. Ventana Registrar Pago 179

Figura 16. Ventana Detalles de Pago 179

Figura 17. Reporte de pagos “ExportarPDF” 180

Figura 18. Reporte de Incidencias “ExportarPDF” 180

Figura 19. Ventana Reportar Incidencia 180

Figura 20. Ventana Detalles de Incidencia 181

Figura 21. Interfaz Principal de inmuebles disponibles 182

Figura 22. Ventana Editar Inmueble 182

Figura 23. Interfaz Registro de Inmueble 183

Figura 24. Interfaz Registro de inmueble “Añadir elementos” 183

Figura 25. Interfaz de vinculación 184

Figura 26. Interfaz de vinculación “Subida de Contrato” 184

Figura 27. Interfaz de gestión de cobros 185

Figura 28. Ventana Validación de Pago 185

Figura 29. Interfaz de gestión de Incidencias 186

Figura 30. Ventana Detalle de Incidencia 186

Figura 31. Interfaz reporte General 187

Figura 32. Interfaz principal Usuario de control parental 187

INTRODUCCIÓN

El incremento en la demanda de alojamiento para la comunidad estudiantil de la Universidad Estatal de Bolívar plantea la necesidad de optimizar los métodos tradicionales de arrendamiento. El sistema web “GestInWEB” se diseñó como una herramienta informática para centralizar, organizar y automatizar los procesos operativos en los inmuebles de alquiler ubicados en la parroquia Guanujo, cantón Guaranda.

Esta plataforma unifica las tareas administrativas, el control económico y la comunicación de problemas de infraestructura en un solo entorno digital. Con esto se busca reducir el papeleo, evitar malentendidos en las cuentas mensuales y agilizar la atención a los daños o reparaciones que soliciten los inquilinos.

Objetivo

Guiar al usuario en el uso correcto de las interfaces y módulos que componen el sistema “GestInWEB”.

Alcance

Este documento describe las funciones accesibles desde el entorno web del sistema, diferenciadas según los privilegios asignados a cada rol de usuario. El alcance contempla los siguientes apartados:

Módulo administrativo: Edición de los respectivos perfiles, registrar inmuebles, vincular, renovación y finalización de contratos de arrendamiento.

Módulo financiero: Registro de los pagos mensuales por concepto de alquiler, control de pagos por inmueble y la generación de comprobantes y reportes descargables en formato PDF.

Módulo de incidencias: Sistema de tickets para que los estudiantes reporten daños materiales en las habitaciones, permitiendo al administrador dar seguimiento al estado de la reparación (Pendiente, En Proceso, Resuelto).

El manual cubre desde el proceso de autenticación (inicio de sesión) hasta la salida segura de la plataforma, limitándose exclusivamente a las funciones programadas dentro de la aplicación web.

Restricciones

El uso y funcionamiento del sistema "GestInWEB" está sujeto a las siguientes limitaciones tecnológicas y operativas:

Dependencia de conectividad: Al ser una aplicación basada en la web, el sistema requiere una conexión estable a internet. No cuenta con un modo de trabajo fuera de línea (offline).

Compatibilidad de navegadores: La interfaz está optimizada para funcionar en versiones actualizadas de Google Chrome, Mozilla Firefox, Microsoft Edge y Safari. Su rendimiento o visualización puede variar en navegadores obsoletos o no comerciales.

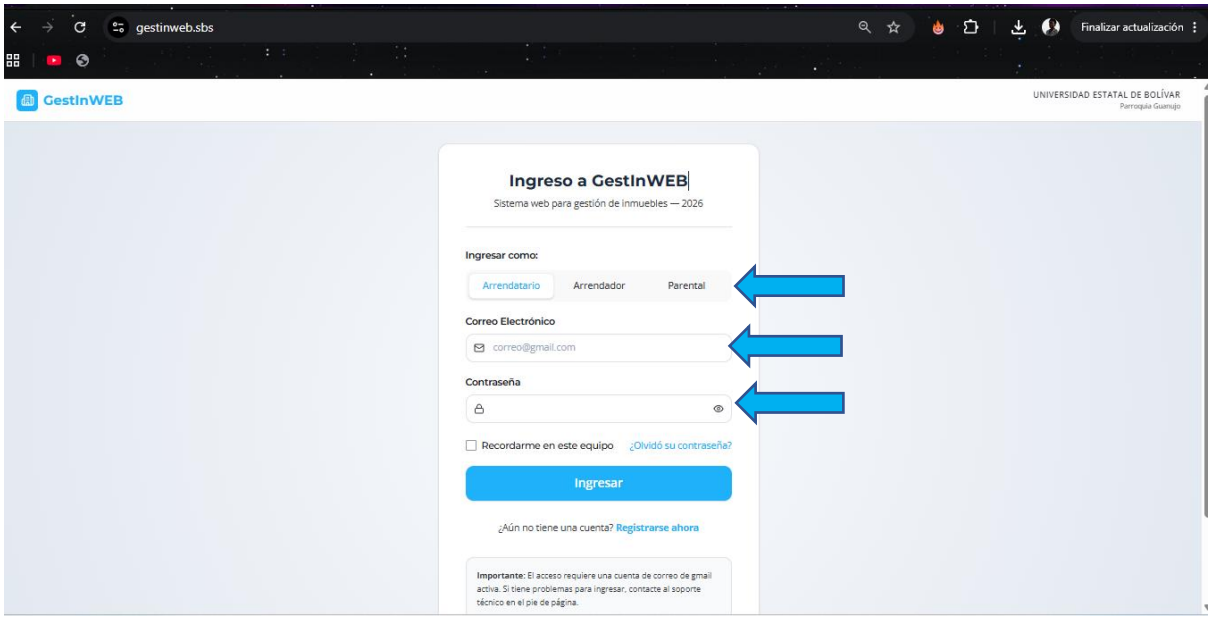
Pagos no automatizados: El sistema funciona como un registro contable y administrativo de los flujos de dinero. No realiza transacciones bancarias directas, transferencias interbancarias ni cobros automáticos mediante pasarelas de pago (tarjetas de crédito/débito). El registro del pago depende de la validación física o manual del administrador.

INGRESO AL SISTEMA

Login y página principal

Para ingresar al sistema puede utilizar cualquier navegador de su preferencia o se sugiere utilizar Google Chrome 145 ingresando la dirección de enlace https://www.gestinweb.sbs a continuación se observa la interfaz principal de ingreso al sistema:

Interfaz inicial del sistema



Nota. Elaborado por: Andrade P. Kevin (2026).

Para ingreso al sistema se debe ingresar con el correo y contraseña que inicialmente se usó para el registro, además de elegir a que rol se va a ingresar según lo que seleccionó al momento del registro.

Recuperar contraseña

También está la interfaz de recuperación de contraseña, en caso de que un usuario se olvide la contraseña, puede seleccionar el enlace de “Olvidó su contraseña”.

Interfaz recuperar contraseña



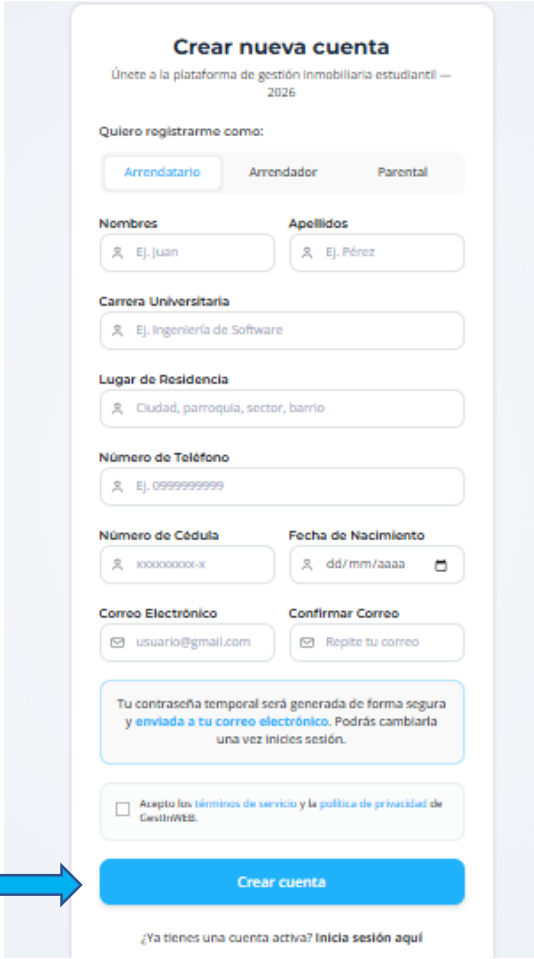
Nota. Elaborado por: Andrade P. Kevin (2026).

El usuario ingresa el correo, el sistema valida el correo y envía una contraseña temporal al correo del usuario, donde tendrá que ingresar esa contraseña en el login para ingresar al sistema el cual este mismo obliga a cambiarla de inmediato.

Registro de Usuario

En esta interfaz el usuario se registra, este formulario tiene tres roles para registrarse, pero sin embargo tiene que registrarse al con el que permitirá gestionar el inmueble. El correo que ingrese al momento del registro tiene que ser valido, debido a que se enviará un enlace de con la contraseña temporal para el ingreso por primera vez a la plataforma:

Interfaz Registro

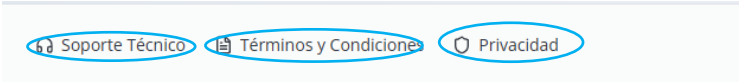


Nota. Elaborado por: Andrade P. Kevin (2026).

Soporte, Términos, y políticas de privacidad

Al pie de la página de todas las interfaces se encuentran estos tres apartados:

Pie de pagina de la plataforma



Nota. Elaborado por: Andrade P. Kevin (2026).

Soporte técnico: En caso de algún problema con la plataforma los datos de contacto para el soporte técnico se encuentran disponibles, hay que tener en cuenta los horarios de atención:

Ventana de soporte técnico



Nota. Elaborado por: Andrade P. Kevin (2026).

Términos y condiciones: Aquí se especifican las reglas de uso, obligaciones del usuario, limitaciones de responsabilidad y modificaciones de términos:

Ventana de Términos y condiciones



Nota. Elaborado por: Andrade P. Kevin (2026).

Políticas de privacidad: Aquí se especifica la información que se recopila, cómo se utiliza esa información y la seguridad de los datos:

Ventana de Políticas de seguridad

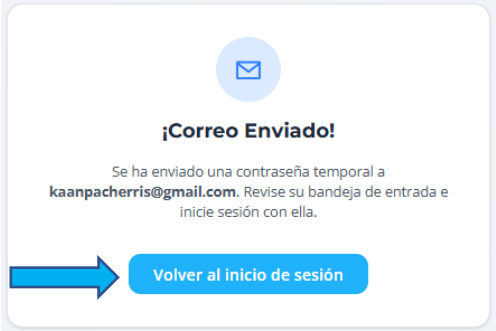


Nota. Elaborado por: Andrade P. Kevin (2026).

Cambio de contraseña Obligatoria

Para recuperación de contraseña o ingreso por primera vez a la plataforma, el sistema le genera una clave temporal y enviada al correo. Para volver al login presionar el botón “volver al inicio de sesión”:

Ventana de correo enviado con exito



Nota. Elaborado por: Andrade P. Kevin (2026).

Ingreso al sistema con la contraseña temporal

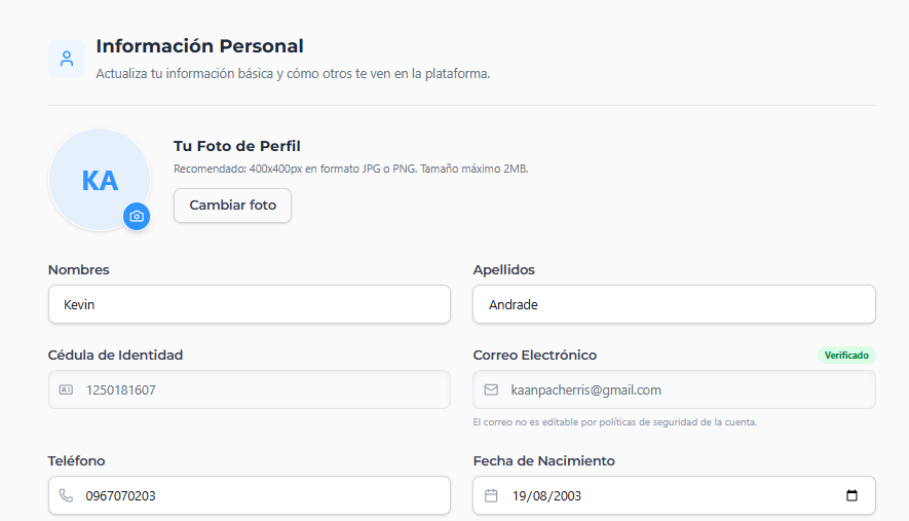


Nota. Elaborado por: Andrade P. Kevin (2026).

Editar perfil

Al ingresar al módulo de arrendatarios podrán acceder directamente a la interfaz de edición de perfil, lo cual permite cambiar todos los datos a excepción de la cedula, fecha de nacimiento y el correo electrónico, también podrán añadir una foto para el perfil, el cual observarán en todo momento al navegar por los modulos:

Interfaz de configuración de Perfil



Nota. Elaborado por: Andrade P. Kevin (2026).

Interfaz de configuración de Perfil "Fotografía editada"



Nota. Elaborado por: Andrade P. Kevin (2026).

PROCESOS MODULO ARRENDATARIOS

Interfaz de la pantalla inicial del arrendatario

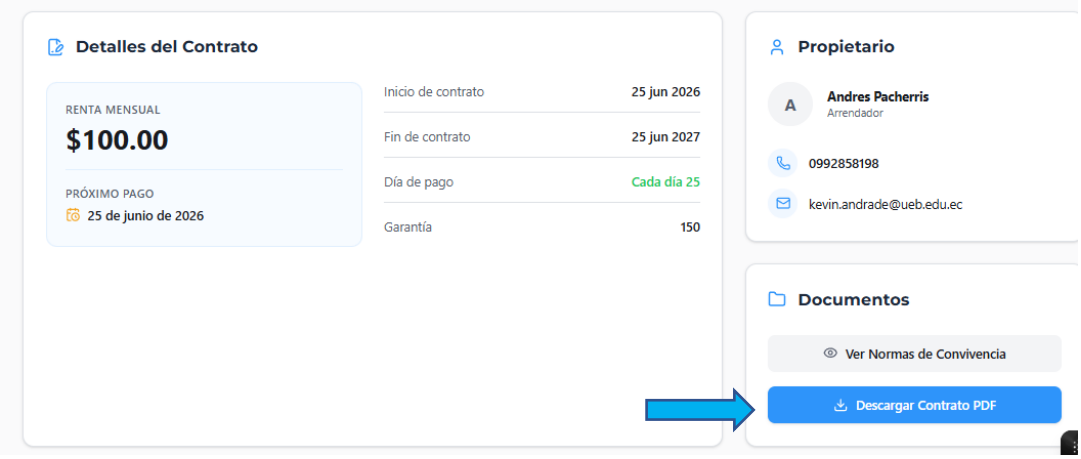
En esta interfaz se observa la información respectiva del inmueble junto con los servicios que cuenta, también se puede ver los detalles del contrato, los datos del arrendador, ver las normas de convivencia y también se puede descargar el contrato presionando el botón "Descargar el contrato PDF".

Interfaz Principal del arrendatario "vizualización inmueble"



Nota. Elaborado por: Andrade P. Kevin (2026).

Interfaz Principal del arrendatario "Visualización contrato"

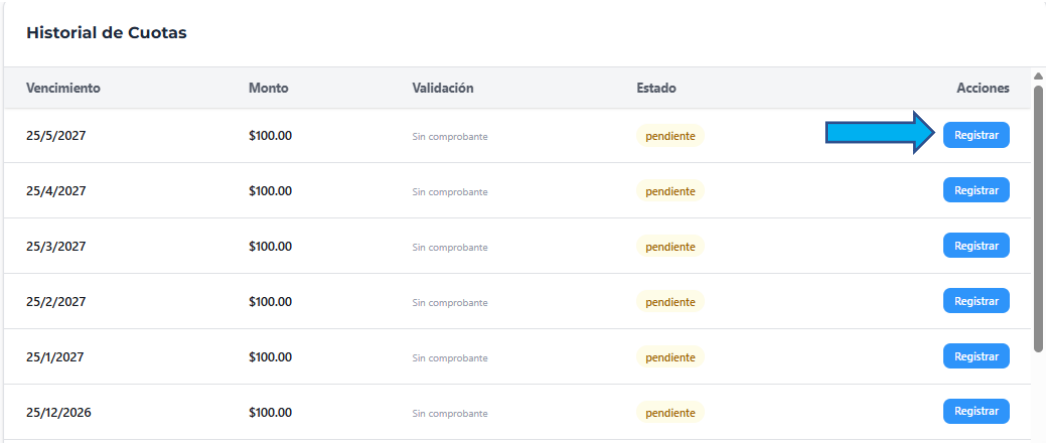


Nota. Elaborado por: Andrade P. Kevin (2026).

Interfaz del visualización y registro de Pagos

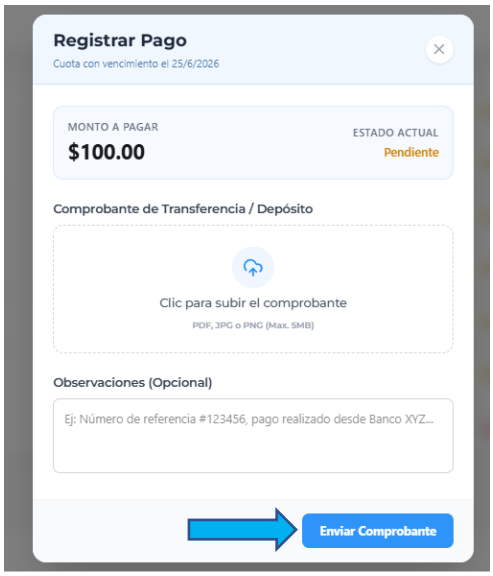
El arrendatario observa el historial de fechas de los pagos mensuales que debe realizar y hace el respectivo registro de pago según la fecha que corresponde presionando el botón "Registrar"

Interfaz del visualización y registro de Pagos



Nota. Elaborado por: Andrade P. Kevin (2026).

Ventana Registrar Pago



Nota. Elaborado por: Andrade P. Kevin (2026).

Al momento de que el arrendatario hace el pago, este pasa a un estado por aprobar, hasta que el arrendador apruebe el pago se podrá editar el pago en caso de algún error, ya cuando se apruebe el pago el arrendatario recibe el comprobante por correo:

Ventana Detalles de Pago

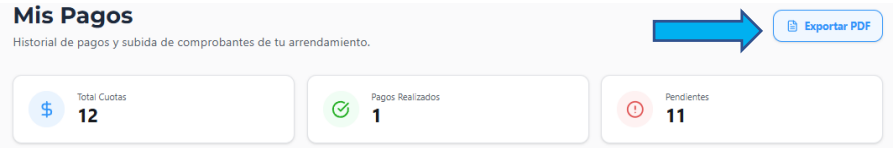


Nota. Elaborado por: Andrade P. Kevin (2026).

Reportes PDF

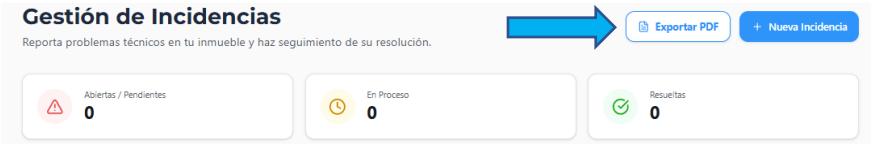
Para generar los reportes en un documento pdf se presiona el botón de “Exportar PDF”

Reporte de pagos “ExportarPDF”



Nota. Elaborado por: Andrade P. Kevin (2026).

Reporte de Incidencias “ExportarPDF”



Nota. Elaborado por: Andrade P. Kevin (2026).

Interfaz para reportar incidencia

Aquí el arrendatario realiza el reporte de incidencias ya se por mantenimiento, daños en el inmueble o problemas con otros inquilinos, donde se detalla a la incidencia con un título, descripción, prioridad y una evidencia fotográfica. Para reportar la incidencia presionar el botón “Enviar reporte”

Ventana Reportar Incidencia



Nota. Elaborado por: Andrade P. Kevin (2026).

Cuando el arrendador resuelve la incidencia el estado pasa a “Resuelto”, con una descripción y evidencia fotográfica de lo que hizo el dueño del inmueble para arreglar el asunto, para editar la incidencia se presiona el botón de “Editar” que está en la parte superior derecha de la ventana:

Ventana Detalles de Incidencia



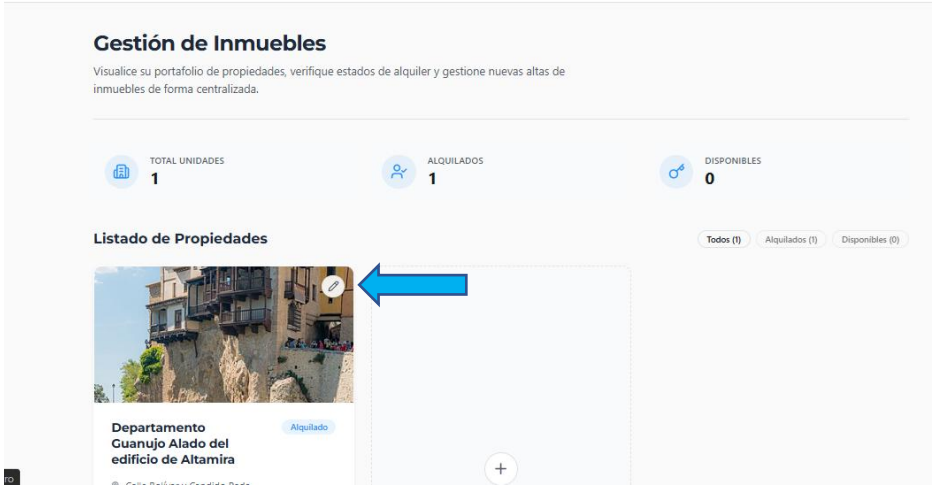
Nota. Elaborado por: Andrade P. Kevin (2026).

PROCESOS MODULO ARRENDADORES

Interfaz Principal de inmuebles disponibles

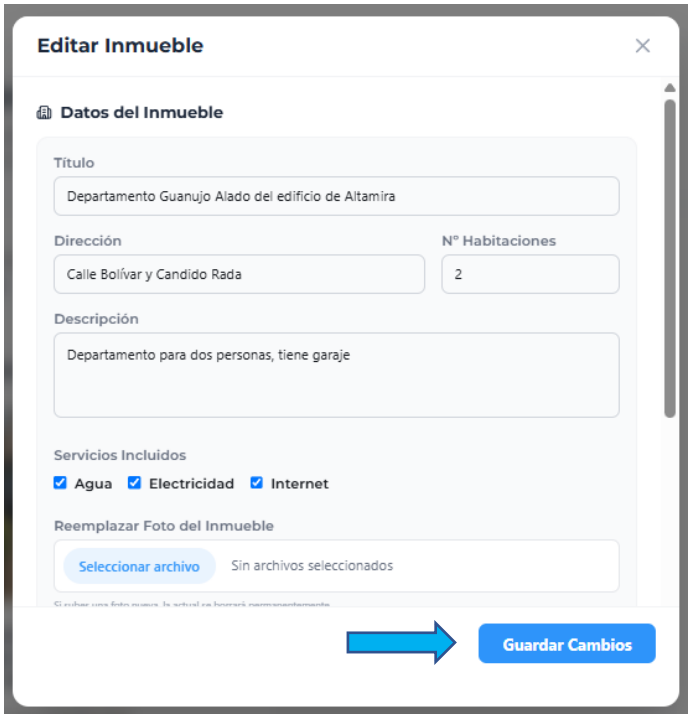
En este módulo se visualizan todos los inmuebles los alquilados y los disponibles también se va a poder ver la información de los inquilinos, detalles del inmueble, e inclusive se puede editar esta información, seleccionando la plumita que está a la parte derecha superior del inmueble. Para guardar los cambios que se editó se presiona el botón “Guardar Cambios”:

Interfaz Principal de inmuebles disponibles



Nota. Elaborado por: Andrade P. Kevin (2026).

Ventana Editar Inmueble

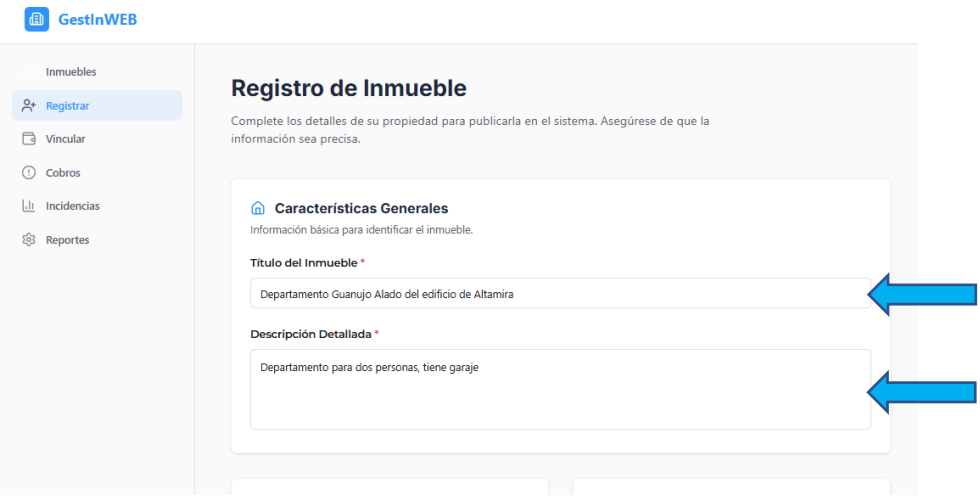


Nota. Elaborado por: Andrade P. Kevin (2026).

Interfaz de registro de Inmueble

En esta pantalla se registran los inmuebles del arrendador, donde se introducen datos como, nombre del inmueble, características, descripción, añadir elementos (se puede añadir elementos en caso de que vengan con el inmueble), y servicios básicos (luz, agua, internet).

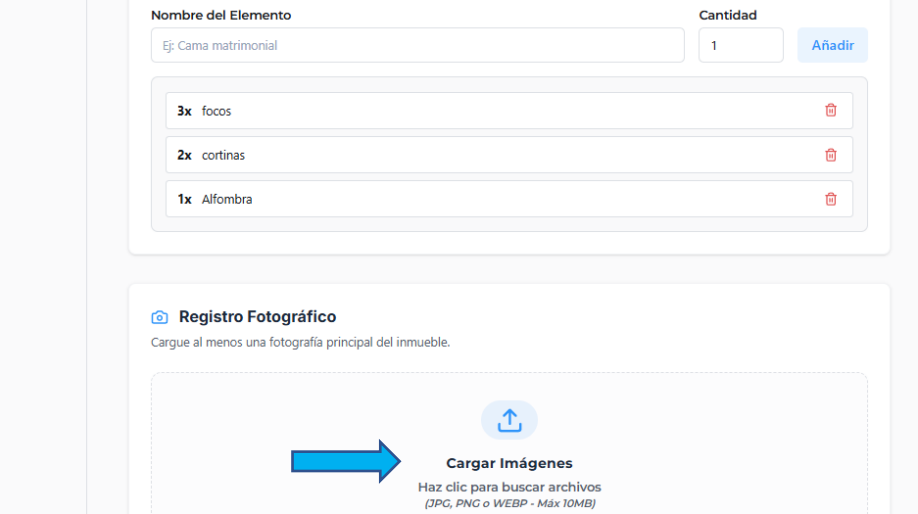
Interfaz Registro de Inmueble



Nota. Elaborado por: Andrade P. Kevin (2026).

Para subir una fotografía también se puede arrastrar el archivo hacia la parte de "Cargar imagen":

Interfaz Registro de inmueble "Añadir elementos"



Nota. Elaborado por: Andrade P. Kevin (2026).

Interfaz de vinculación de partes involucradas en el contrato

En esta interfaz es donde se formaliza el contrato y se vincula a los arrendatarios y usuarios de control parental con el inmueble permitiendo la visualización de este en las interfaces de sus

respectivos roles, en el caso del usuario de control parental es opcional y/o dependerá de cómo estará estipulado el contrato.

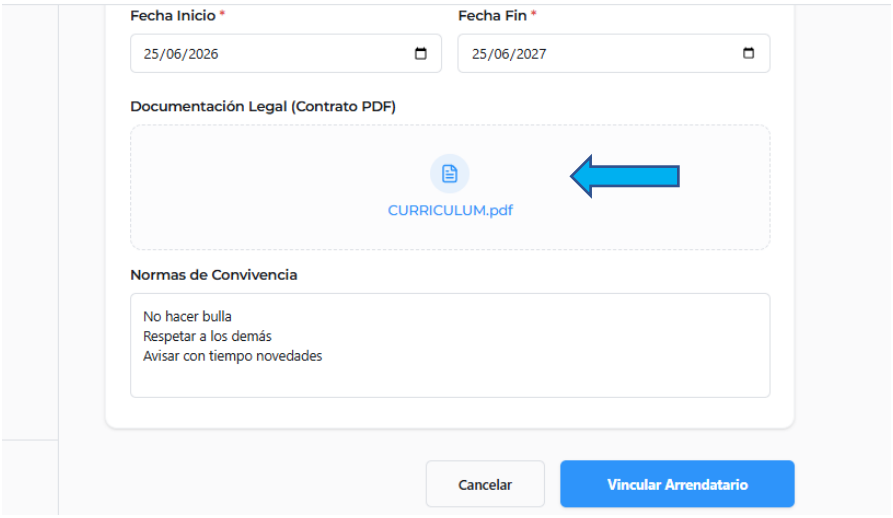
Interfaz de vinculación



Nota. Elaborado por: Andrade P. Kevin (2026).

Se arrastra el documento hacia la parte de “Contrato PDF”:

Interfaz de vinculación “Subida de Contrato”

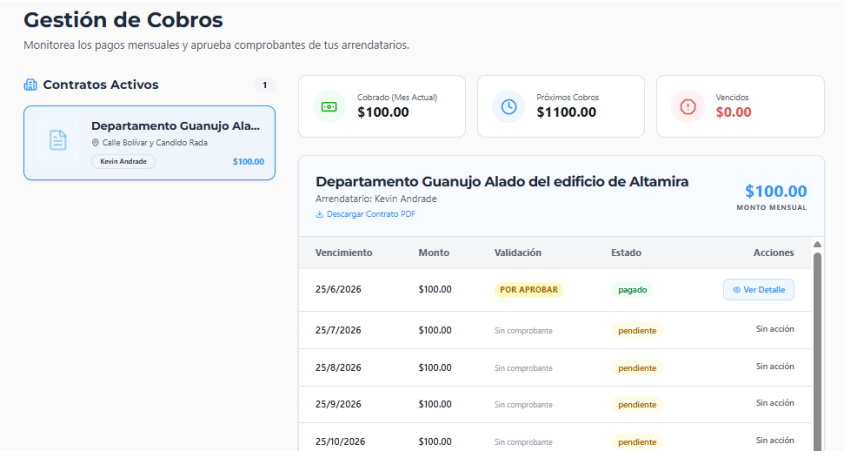


Nota. Elaborado por: Andrade P. Kevin (2026).

Interfaz de gestión de cobros

En esta interfaz se observan los inmuebles con sus respectivas fechas de pagos, también hace una contabilidad total de los ingresos y pagos pendientes.

Interfaz de gestión de cobros

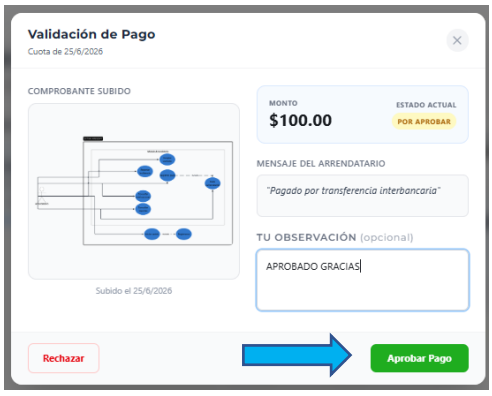


Nota. Elaborado por: Andrade P. Kevin (2026).

Validación de Pago

Al momento que el arrendatario realiza un pago, el arrendador debe validarlo, seleccionando el botón de “ver detalle”, desplegándose una ventana indicando los detalles del pago con una evidencia fotográfica, el arrendador al verificar que el pago se hizo correctamente da paso a la aprobación del pago dejando un comentario, también a su vez se genera un comprobante automáticamente que se envía al correo del arrendatario

Ventana Validación de Pago



Nota. Elaborado por: Andrade P. Kevin (2026).

Interfaz de gestión de Incidencias

Aquí se visualizan las incidencias hechas por los inquilinos, estas se separan según el inmueble, y para darle solución a una incidencia debe seleccionar en el botón “ver/Resolver” abriéndose una ventana donde se ve la información de la incidencia, cambiando el estado de (pendiente a resuelto) dejando una descripción de lo que se hizo, y añadiendo una evidencia fotográfica.

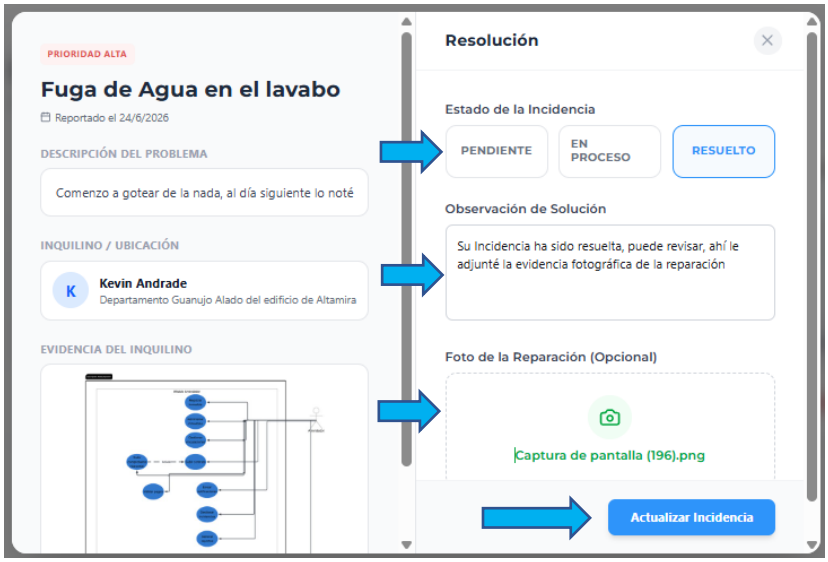
Interfaz de gestión de Incidencias



Nota. Elaborado por: Andrade P. Kevin (2026).

En caso de actualizar la resolución de la incidencia del arrendatario, se editan los datos correspondientes y se presiona el botón de “Actualizar Incidencia”:

Ventana Detalle de Incidencia

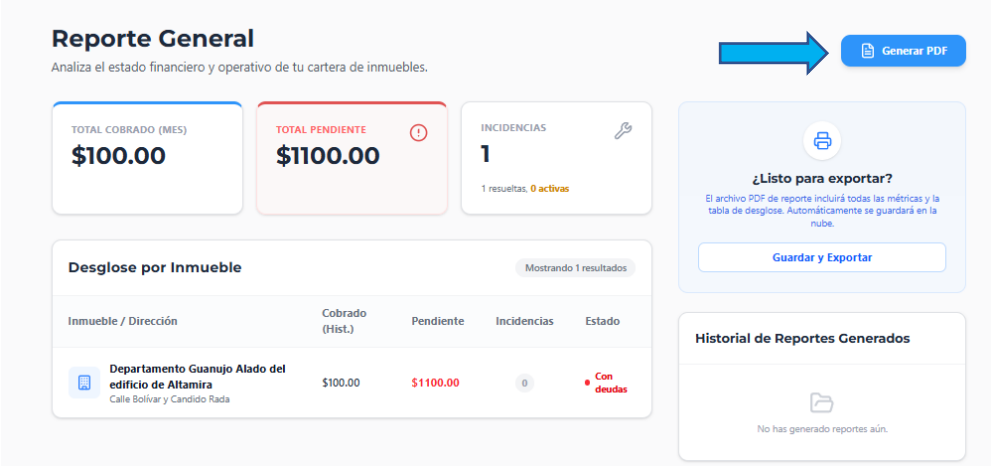


Nota. Elaborado por: Andrade P. Kevin (2026).

Interfaz reporte General

En este módulo el arrendadore puede realizar reportes pdf de los pagos por inmueble, presionando el botón de “Generar PDF”

Interfaz reporte General



Nota. Elaborado por: Andrade P. Kevin (2026).

PROCESOS MODULO USUARIO CONTROL PARENTAL

Interfaz principal Usuario de control parental

En este módulo el usuario solo tiene permisos de lectura, es decir no puede eliminar, o actualizar algún registro, debido a que este usuario solo esta como supervisor del estudiante que alquila el inmueble.

Interfaz principal Usuario de control parental



Nota. Elaborado por: Andrade P. Kevin (2026).

Anexo 5

Certificado Anti plagio

Certificado Anti plagio

UEB
UNIVERSIDAD ESTATAL DE BOLIVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

Dr. HENRY FERNANDO VALLEJO BALLESTEROS EN CALIDAD DE DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR,

CERTIFICA

Que el trabajo de integración curricular denominado "**SISTEMA WEB INTEGRAL PARA LA GESTIÓN ADMINISTRATIVA, FINANCIERA Y DE INCIDENCIAS EN INMUEBLES DE ALQUILER PARA ESTUDIANTES DE LA UNIVERSIDAD ESTATAL DE BOLÍVAR, PARROQUIA GUANUJO, 2026.**", presentado por Kevin Andres Andrade Pacherris estudiante de la **carrera de Software** pasó el análisis de coincidencia no accidental en la herramienta COMPILATIO, reflejando un **porcentaje de similitud del 10%**, como se puede evidenciar en el documento adjunto.

Guaranda, 02 de Julio del 2026

Atentamente,

Validez únicamente en Fírmalo.
Firmado electrónicamente por:
HENRY FERNANDO
VALLEJO BALLESTEROS

Dr. Henry Vallejo
Director

Dirección: Av. Ernesto Che Guevara y Gabriel Secaira
Guaranda-Ecuador
Teléfono: (593) 3220 6059
www.ueb.edu.ec

Certificado de análisis
Compilatio Magister+ | UNIVERSIDAD ESTATAL DE BOLIVAR (UEB)

SISTEMA WEB INTEGRAL PARA LA GESTIÓN ADMINISTRATIVA, FINANCIERA Y DE INCIDENCIAS EN INMUEBLES DE ALQUILER PARA ESTUDIANTES DE LA UNIVERSIDAD ESTATAL DE BOLÍVAR, PARROQUIA GUANUJO, 2026.

ID : 4a8d7db240b78470fc716c0197cd01888f41528d

10% Textos sospechosos

Nombre del fichero : SISTEMA WEB INTEGRAL PARA LA GESTIÓN ADMINISTRATIVA, FINANCIERA Y DE INCIDENCIAS EN INMUEBLES DE ALQUILER PARA ESTUDIANTES DE LA UNIVERSIDAD ESTATAL DE BOLÍVAR, PARROQUIA GUANUJO, 2026..txt
Tamaño del archivo original : 15,69 MB
Número de palabras : 28.938

Depositante : HENRY FERNANDO VALLEJO BALLESTEROS
Fecha de depósito : 2 de julio de 2026
Tipo de carga : interface
fecha de fin de análisis : 2 de julio de 2026

Resumen (sección 1/2)

Localización de los textos sospechosos en el documento :



Incluido en el porcentaje de textos sospechosos :

Similitudes 1%

Pasajes con similitudes a fuentes encontradas en diferentes colecciones.



Detección de IA 10%

Textos estilísticamente próximos a un texto generado por una IA.
Este índice es un indicador y no una prueba. Comprueba con el autor si domina los conocimientos mencionados en el documento.



Idiomas no reconocidos 4%

Pasajes en los que parte del vocabulario utilizado no forma parte del diccionario de la lengua.
Puede tratarse de un intento del autor de modificar el texto para evitar ser detectado.



No incluido en el porcentaje de textos sospechosos :

Textos entre comillas 1%

Pasajes entre comillas, a menudo indicativos de una cita.