

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



**“PLATAFORMA PARA LA ADMINISTRACIÓN DE BIENES
RAÍCES QUE BUSCA GENERAR LA TRANSFORMACIÓN
QUE EL RUBRO INMOBILIARIO NECESITA”**

BERNABÉ ANTONIO GARCÍA SOLAR

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA**

Profesor Guía: Pedro Francisco Godoy Barrera
Profesor Correferente: Dayana Carrillo Daboin

Agosto - 2023

RESUMEN

En la última década, se ha observado un crecimiento significativo en la adopción de tecnologías de la información en diversos aspectos de nuestra vida, abarcando desde las actividades cotidianas de las personas hasta el funcionamiento de las grandes empresas. Esta situación ha motivado a las pequeñas, medianas o grandes empresas, a modernizar sus operaciones e incorporar lo que se conoce como la "transformación digital".

En este trabajo, se define y describe una solución destinada a incursionar en el sector de bienes raíces, enfocándose específicamente en el ámbito del arriendo y la administración de propiedades. En los primeros apartados del documento, se exponen las principales problemáticas que afectan a los administradores de propiedades. Luego, se define la implementación de un nuevo software que soluciona varios de los procesos que afrontan los administradores de propiedades en su labor.

Al final del documento se muestran las evidencias que permiten afirmar que la solución desarrollada cumple con los objetivos de ser una plataforma intuitiva y de fácil uso, que permite la centralización de la información y que por medio de ella se logra generar un estado de caja claro y controlado

Palabras clave: Software; transformación digital; propiedades; arriendo; administración.

ABSTRACT

Over the past decade, a significant growth in the adoption of information technologies has been observed in various aspects of our lives, ranging from people's daily activities to the operations of large enterprises. This situation has motivated small, medium, and large companies to modernize their operations and incorporate what is known as "digital transformation".

This paper defines and describes a solution aimed at entering the real estate sector, with a specific focus on the rental and property management domain. The initial sections of the document outline the primary challenges that property managers face. Next, the implementation of new software is defined to address several processes encountered by property managers in their work.

At the end of the document, the evidence is presented that allows one to affirm that the developed solution meets the objectives of being an intuitive and user-friendly platform, enabling the centralization of information, and through it, achieving a clear and controlled cash flow state.

Keywords: Software; digital transformation; properties; rental; management.

GLOSARIO

- AFP: Administradora de Fondos de Pensiones.
- BPMN: Business Process Model and Notation / Notación de Procesos de Negocio
- CBRS: Conservador de Bienes Raíces de Santiago.
- Formulario N° 1: Es la solicitud de regularización para realizar la inscripción de una propiedad en bienes raíces.
- Formulario N° 2.890: Es el formulario por el cual los notarios y conservadores de bienes raíces deben llevar a cabo el traspaso de información al SII.
- PYMES: Pequeñas y Medianas Empresas.
- UTM: Unidad Tributaria Mensual.

ÍNDICE DE CONTENIDOS

RESUMEN	2
ABSTRACT	2
ÍNDICE DE FIGURAS Y TABLAS	6
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	10
1.2 Procesos de la administración inmobiliaria	11
1.2.1 Proceso de arriendo de una propiedad	11
1.2.2 Proceso de pago de cuentas	14
1.2.3 Proceso mantenimiento de propiedades	15
1.2.4 Proceso de registro y aviso de pago de arriendo	16
1.2.5 Proceso de generación del estado de cuenta de una propiedad	18
1.3 Problemáticas generales	18
1.4 Problemáticas específicas del cliente.	21
1.5 Mercado actual.	22
1.5.1 Mercado Chileno	22
1.5.2 Mercado internacional.	23
1.6 Objetivos	24
1.6.1 Objetivo general	24
1.6.2 Objetivos específicos	24
1.7 Alcances y desarrollo de la memoria	25
CAPÍTULO 2: MARCO CONCEPTUAL.	26
2.1 Transformación digital	26
2.1.1 El proceso de Arriendo de una Propiedad	26
2.1.2 Proceso de pago de cuentas	26
2.1.3 El proceso mantenimiento de propiedades	27
2.1.4 Proceso de registro y aviso de pago de arriendo	27
2.1.5 Proceso de generación del estado de cuenta de una propiedad	27
2.2 Metodologías de desarrollo de software	27
2.2.1 Metodologías tradicionales	27
2.2.2 Metodologías ágiles	28
2.2.3 Metodología programación extrema (XP)	30
2.3 Arquitectura de software.	32
2.4 Frontend	38
2.4.1 Herramientas de desarrollo web	38
2.4.2 Librerías JavaScript	41
2.4.3 User Experience (UX)	41

2.5 Backend	43
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	47
3.1 Implementación de la metodología Programación Extrema	47
3.2 Work Breakdown Structure (WBS)	49
3.3 Actores del sistema	51
3.4 Historias de usuario	52
3.5 Base de datos	68
3.5.1 Propietario	70
3.5.2 Propiedad	71
3.5.3 Labor	72
3.5.4 Maestro	74
3.5.5 Gastos	75
3.5.6 Contratos	76
3.5.7 Ingresos	78
3.5.8 Postulante	79
3.5.9 Liquidación	80
3.5.10 Administración	82
3.6 Diseño y usabilidad	83
3.6.1 Colores	85
3.6.2 Diseño de las interfaces gráficas.	85
3.6.3 Evaluación de la usabilidad de un software	92
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN.	96
● Testing de Usuarios	99
● Encuestas de Usabilidad	107
CAPÍTULO 5: CONCLUSIONES.	111
REFERENCIAS BIBLIOGRÁFICAS	113

ÍNDICE DE FIGURAS Y TABLAS

Figura 1. Diagrama BPMN que refleja el proceso inicial de arrendamiento de una propiedad. Fuente: elaboración propia	12
Figura 2. Diagrama BPMN que refleja el proceso una vez finalizado el contrato o acuerdo entre las partes. Fuente: elaboración propia	13
Figura 3. Diagrama BPMN que refleja el proceso de pago de cuentas. Fuente: elaboración propia	14
Figura 4. Diagrama BPMN que refleja el proceso de mantenimiento de propiedades. Fuente: Elaboración Propia	15
Figura 5. Diagrama BPMN que refleja el proceso de pago de arriendo. Fuente: elaboración propia	17
Tabla 1. Comparativa entre los tipos de software que están presentes en el mercado chileno y EasyKey. Fuente: elaboración propia	23
Figura 7. Elaborada por Sinnaps, Diagrama de flujo de la metodología XP Fuente: https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp	31
Figura 8. Esquema de patrón MVC Fuente: https://developer.apple.com/library/archive/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html	36
Figura 9. Elaborado por The Hertz Corporation, Página web de la empresa Hertz Fuente: https://www.hertz.es/rentacar/reservacion/	42
Figura 10. Elaborado por Milka, Página web de la empresa Milka Fuente: https://milka.cl/	43
Figura 11. Elaborado por Zara, Página web de la empresa Zara Fuente: https://www.zara.com/cl/	43
Figura 12. WBS del desarrollo de la aplicación EasyKey generado sobre una metodología XP Fuente: elaboración propia	49
Figura 13: Diagrama de ER del sistema Easy Key Fuente: elaboración propia.	68
Figura 14. Diagrama ER que detalla la macro entidad “Propietarios” y se destaca en rojo la relaciones con las entidad principal de las otras macro entidades. Fuente: elaboración propia	70
Figura 15. Diagrama ER de la primera parte del detalle la macro entidad “Propiedad” y se destaca en rojo las relaciones con la entidad principal de las otras macro entidades. Fuente: elaboración propia	71
Figura 16. Diagrama ER de la segunda parte del detalle la macro entidad “Propiedad” y se destacan en rojo las relaciones con las entidades principales de las otras macro entidades.	

Fuente: elaboración propia	72
Figura 17. Diagrama ER del detalle la macro entidad “Labor” y se destacan en rojo las relaciones con las entidades principales de las otras macro entidades.	
Fuente: elaboración propia	73
Figura 18. Detalle de la macro entidad “Maestro” donde se reflejan en rojo las relaciones principales y las otras relaciones que la complementan.	
Fuente: elaboración propia	74
Figura 19. Detalle de la macro entidad “Gastos” donde se reflejan en rojo las relaciones principales y las otras relaciones que la complementan.	
Fuente: elaboración propia	75
Figura 20. Diagrama ER de la primera parte del detalle la macro entidad “Contratos” y se destaca en rojo las relaciones con la entidad principal de las otras macro entidades.	
Fuente: elaboración propia	76
Figura 21. Diagrama ER de la segunda parte del detalle la macro entidad “Contratos” y se destaca en rojo las relaciones con la entidad principal de las otras macro entidades.	
Fuente: elaboración propia	77
Figura 22. Detalle de la macro entidad “Ingresos” donde se reflejan en rojo las relaciones principales y las otras relaciones que la complementan.	
Fuente: elaboración propia	78
Figura 23. Detalle de la macro entidad “Postulante” donde se reflejan en rojo las relaciones principales y las otras relaciones que la complementan.	
Fuente: elaboración propia	80
Figura 24. Primera parte del detalle de la macro entidad “Liquidación” donde se reflejan en rojo las relaciones principales y las otras relaciones que la complementan.	
Fuente: elaboración propia	81
Figura 25. Diagrama ER de la segunda parte del detalle de la macro entidad “Liquidación” y se destaca en rojo las relaciones con la entidad principal de las otras macro entidades.	
Fuente: elaboración propia	81
Figura 26. Detalle de la macro entidad “Administración” donde se reflejan en rojo las relaciones principales y las otras relaciones que la complementan.	
Fuente: elaboración propia	82
Figura 27. Paleta de colores principales de la aplicación EasyKey y su logo.	
Fuente: elaboración propia	85
Figura 28. Diseño de menú principal panel de administración de Easy Key.	
Fuente: elaboración propia	86
Figura 29. Formulario de creación de propietarios de Easy Key.	
Fuente: elaboración propia	87
Figura 30. Error en el formulario de creación de propietarios de Easy Key.	
Fuente: elaboración propia	88
Figura 31. Formulario de creación de propietarios después del error.	
Fuente: elaboración propia	89
Figura 32. Proceso de creación de propietario exitoso.	

Fuente: elaboración propia	90
Figura 33. Pantalla del detalle de un propietario. Fuente: elaboración propia	90
Figura 34. Pantalla del detalle de un propietario. Fuente: elaboración propia	91
Tabla 2. Listado de sentencias obtenidas del cuestionario CSUQ y ASQ que evalúan la usabilidad de un sistema. Fuente: elaboración propia	95
Figura 35. Estado de cuenta de la propiedad Agustinas 813 Of. 809 Fuente: elaboración propia	96
Figura 36. Estado de cuenta de la propietaria Rosa Solar. Fuente: elaboración propia	97
Figura 37. Liquidación enviada a la propietaria Rosa Solar. Fuente: elaboración propia	97
Figura 38. Liquidación que envía la empresa Arrecife a sus socios. Fuente. Elaboración empresa Arrecife	98
Tabla 3. Usuarios que participaron en el testing de la aplicación Easy Key Fuente: elaboración propia	99
Tabla 4. Tiempos promedios de la labor “Crear un propietario” y la cantidad de casos exitosos, separados por la experiencia inmobiliaria. Fuente: elaboración propia.	100
Tabla 5. Tiempos promedios de la labor “Crear un propietario” y la cantidad de casos exitosos, separados por la experiencia en tecnologías similares. Fuente: elaboración propia.	100
Tabla 6. Tiempos promedios de la labor “Crear una propiedad” y la cantidad de casos exitosos, separados por la experiencia inmobiliaria. Fuente: elaboración propia.	101
Tabla 7. Tiempos promedios de la labor “Crear una propiedad” y la cantidad de casos exitosos, separados por la experiencia en tecnologías similares Fuente: elaboración propia.	102
Tabla 8. Tiempos promedios de la labor “Agregar un propietario a la propiedad creada” y la cantidad de casos exitosos, separados por la experiencia inmobiliaria. Fuente: elaboración propia	102
Figura 39. Pantalla donde se presenta el problema de la labor “Agregar un propietario a la propiedad creada” Fuente: elaboración propia	103
Tabla 9. Tiempos promedios de la labor “Agregar un propietario a la propiedad creada” y la cantidad de casos exitosos, separados por la experiencia en tecnologías similares Fuente: elaboración propia	103
Tabla 10. Tiempos promedios de la labor “Agregar un gasto único a una propiedad en el mes actual” y la cantidad de casos exitosos, separados por la experiencia inmobiliaria. Fuente: elaboración propia	104

Tabla 11. Tiempos promedios de la labor “Agregar un gasto único a una propiedad en el mes actual” y la cantidad de casos exitosos, separados por la experiencia en tecnologías similares	
Fuente: elaboración propia	105
Tabla 12. Tiempos promedios de la labor “Agregar un ingreso a la propiedad creada” y la cantidad de casos exitosos, separados por la experiencia inmobiliaria.	
Fuente: elaboración propia	106
Tabla 13: Tiempos promedios de la labor “Agregar un ingreso a la propiedad creada” y la cantidad de casos exitosos, separados por la experiencia en tecnologías similares	
Fuente: elaboración propia	107
Tabla 14: Resultados promedios de la encuesta generada por preguntas de la CSUQ y ASQ post testing de usuario.	
Fuente: elaboración propia	108

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

En Chile, se ha observado a través de los años que el uso de la tecnología se ha incrementado debido a múltiples factores, tales como el aumento en el acceso a internet o la gran cantidad de dinero en circulación debido a los retiros de la AFP, que han favorecido la adquisición de productos tecnológicos (*Mercado Tecnológico En Chile 2020, 2021*). Uno de los catalizadores más relevantes en estos últimos años ha sido la pandemia por el virus COVID-19, que ha generado una transformación drástica tanto en el diario vivir como a nivel industrial. Lo anterior muestra un punto de inflexión entre los años 2020 y 2021 respecto a la era de la presencialidad, provocando una disminución en el trabajo en formato presencial y generando un aumento en el formato remoto. Sin embargo, a medida que ha avanzado el año 2023, las cifras de este último han sufrido una leve disminución, lo cual hace pensar que a medida que pase el tiempo, este mismo puede llegar a desaparecer, sin embargo según los gráficos y la información obtenida de la Cámara de Comercio de Santiago (*Porcentaje De Personas En Teletrabajo Disminuye a Un Tercio Del Empleo – Cámara De Comercio De Santiago – CCS, 2020*), permiten entender y visualizar que en el futuro, el método de trabajo remoto no va a desaparecer, si no, que será complementario al trabajo presencial. Por otro lado según el estudio realizado por la empresa Talana (*Un 72% de las empresas planea mantener algún grado de teletrabajo - Diario El Mostrador, 2022*), un 72% de las empresas que no tenían teletrabajo antes de la pandemia y tuvieron que modificar su forma de trabajar, tienen planeado mantener esta modalidad en algún grado, lo cual indica que el teletrabajo no va a ser una forma de trabajo generada por la necesidad, sino que una modalidad de trabajo más dentro de la empresa. Incluso en Chile en el año 2020 se dictó la Ley N°21.220 que regula el trabajo a distancia, la cual fue introducida como un nuevo capítulo dentro del Código del Trabajo (*Ley 21220, Social, P. T. D. M. Y. - 2020, 26 marzo*).

Muchos profesionales tales como Val Roman de la Universidad de Deusto (*Industria 4.0: La Transformación Digital De La Industria, n.d.*), Juan Ferreira de la Universidad de Málaga (*Industria 4.0 Y Transformación Digital: Nuevas Formas De Organización Del Trabajo, 2019*), entre otros, denominan lo que se vive en la actualidad como la nueva revolución industrial, que apunta directamente a diferentes soluciones que entregan las nuevas tecnologías de software las cuales le permitan a los clientes obtener datos que puedan analizar y trabajar, para mejorar sus decisiones frente a los cambios que se están generando dentro del mercado, explicando que mientras más datos y más información se pueda obtener de cada mercado mayor será la utilidad de la empresa o PYME en la cual se esté implementando.

Debido a los factores antes mencionados, así como a la motivación de obtener mayores utilidades, diferentes empresas y PYMES dedicadas al desarrollo de software, han visto una oportunidad para generar nuevos negocios y nuevas aplicaciones que generen utilidades en diferentes áreas de la industria. Desde algo simple como un agendamiento, una página con venta online, hasta un simple código QR, se han mostrado como una solución y un negocio que permite a las empresas, generar un cambio innovador dentro de

su forma de trabajar (A Un año De La Pandemia: Pymes Chilenas Aceleraron Su Transformación Digital, 2021).

Una industria que se ha visto poco influenciada por esta revolución digital es el área inmobiliaria. Esta situación se debe a diferentes razones, entre ellas se encuentran los marcos legales que tiene el proceso de arriendo de un bien inmobiliario, la resistencia al cambio presentada por los administradores de propiedad, llámese administradores a las personas o empresas que se encargan de administrar una propiedad y que no necesariamente es la propietaria del bien raíz. Actualmente, en el mercado de arriendo de propiedades existen varias aplicaciones que solucionan la tarea particular de encontrar un arriendo y/o comprar un bien raíz, algunas de estas aplicaciones son Airbnb, Toctoc, Portal Inmobiliario, entre otras. Sin embargo, estas soluciones dejan de lado el papel crucial que ejercen los administradores y dueños de estas propiedades, los cuales tienen otras labores que no son abordadas por estas aplicaciones. Algunas de estas tareas son las siguientes:

- Lograr un control de las finanzas del inmueble.
- Lograr un control de los pagos del arrendatario.
- Lograr un control del estado real del inmueble.
- Existe un gran gasto de tiempo a la hora de firmar un contrato con el arrendatario.
- Lograr un control del contrato o escritura entre ambas partes.
- Lograr un respaldo de la comunicación oficial entre ambas partes.
- No tienen las herramientas necesarias para conocer a las personas a las cuales se les vende o arrienda una propiedad.
- Tienen un gran gasto de tiempo en la transferencia de las cuentas básicas entre el arrendatario y el arrendador.

Estas tareas u obligaciones de los administradores y dueños de las propiedades son claves para lograr que el negocio inmobiliario genere una utilidad a corto y a largo plazo, ya que si se logra un control en las finanzas del inmueble, manteniendo las cuentas de la propiedad al día y se realiza un seguimiento en la contabilidad general de la propiedad considerando los ingresos y egresos de cada inmueble, el administrador tendrá los datos necesarios para tomar decisiones sobre el bien raíz y lograr dirimir si la propiedad le genera utilidad o no.

1.2 Procesos de la administración inmobiliaria

Para generar una solución que abarque la administración de propiedades, se deben comprender los procesos de forma independiente, desde el proceso de pago de cuentas, hasta la labor de arriendo de una propiedad, considerando el pago del arriendo y las labores de mantención y/o arreglos de los inmuebles.

1.2.1 Proceso de arriendo de una propiedad

Para generar una solución para el arriendo de una propiedad, es fundamental entender los procesos involucrados en esta labor, identificando cada una de las problemáticas que se encuentran dentro del mismo. Por lo tanto, en la figura 3 se muestra un diagrama BPMN

que refleja el proceso de arriendo de una propiedad. Sin embargo, antes de profundizar en el proceso en sí, es fundamental identificar los actores principales de este proceso, que son tres en total:

- **Arrendador o administrador:** persona u organización, que se encarga de arrendar una propiedad a personas que lo soliciten, puede ser el propietario del bien raíz, un corredor de propiedades o una persona a la cual le dieron la responsabilidad de hacerse cargo de la propiedad.
- **Arrendatario:** persona u organización que adquiere el derecho a usar el inmueble a cambio de un pago.
- **Notario (Opcional):** funcionario público que tiene autoridad para dar fe de los actos públicos realizados ante él, y además redacta y garantiza documentos, como testamentos, contratos, escrituras de compraventa, entre otros.

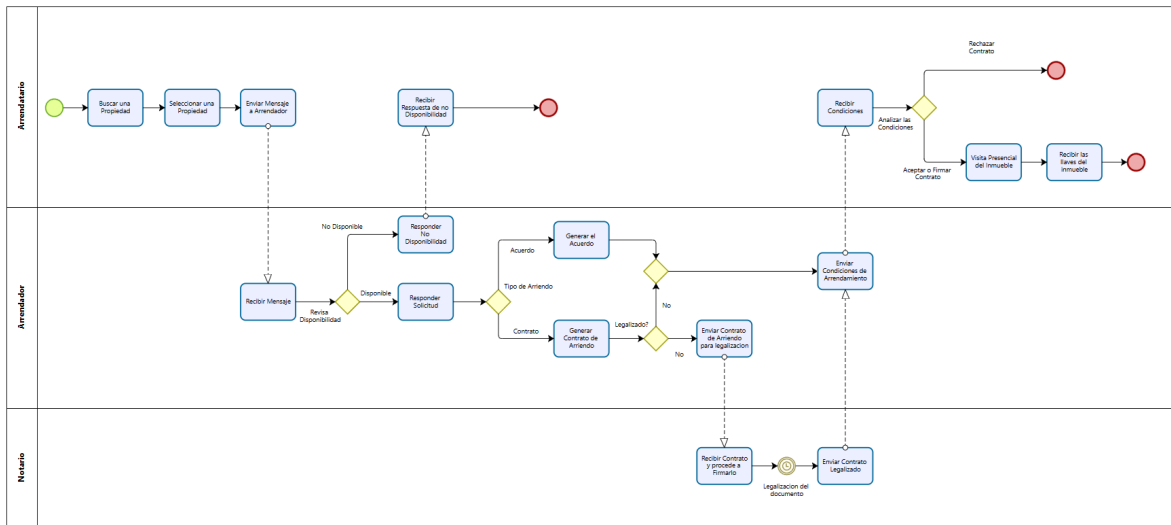


Figura 1. Diagrama BPMN que refleja el proceso inicial de arrendamiento de una propiedad.

Fuente: elaboración propia

Para el inicio del proceso, el arrendatario busca una propiedad para poder hacer uso de este, al encontrarla se encarga de buscar un medio de comunicación con la entidad a cargo del proceso de arriendo, luego el arrendador procede a revisar la disponibilidad de este, en caso de que no se encuentre disponible, procede a informar al arrendatario y rechazar la posibilidad del acuerdo. En caso de que la propiedad esté disponible, se procede a generar un acuerdo entre las partes. En Chile las leyes permiten que el acuerdo sea oral, conocido como acuerdo de palabra, o por medio de un contrato, el cual es un papel físico o digital que es firmado por ambas partes, este puede ser legalizado ante notario en caso de que uno de los actores lo considere necesario.

En caso de que el contrato sea legalizado, este debe ser consultado por un notario público y firmado por el mismo, luego de que el acuerdo se encuentre regulado por la entidad pública procede a ser enviado al posible arrendatario, este mismo analiza las condiciones

que se encuentran en el documento, en el caso de que lo rechace el proceso de arriendo es cerrado y terminado, en cambio en caso de ser aceptado, se gestiona una visita a la propiedad y a firmar el contrato de arrendamiento, finalmente se entrega la propiedad, concluyendo de este modo el proceso inicial de arrendamiento, hasta que el contrato sea terminado o existan faltas en las cláusulas de dicho contrato/acuerdo (*7 Requisitos Para Firmar Un Contrato De Arriendo Notarial*, n.d.).

Continuando, luego de que haya transcurrido la duración del contrato, se procede a realizar el proceso de finalización del contrato/acuerdo entre los protagonistas, en la figura 2 se muestra el proceso de finalización y los posibles problemas que pueden existir al concluir el mismo.

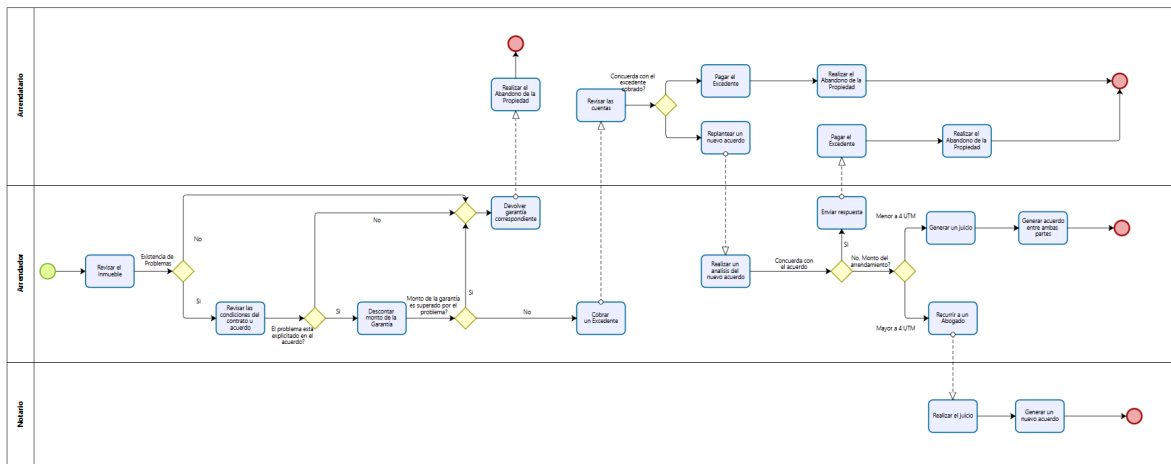


Figura 2. Diagrama BPMN que refleja el proceso una vez finalizado el contrato o acuerdo entre las partes.

Fuente: elaboración propia

Este proceso comienza con la revisión de la propiedad por parte del arrendador, revisando que la propiedad esté en las mismas condiciones en que fue entregada inicialmente. En caso de que el inmueble no tenga problemas, el arrendatario se retira de la propiedad y finaliza el proceso de arriendo. Sin embargo, en caso de que existan diferencias en el estado de la propiedad, antes y después del contrato de arrendamiento, se procede a descontar de la garantía pagada por el arrendatario al inicio del proceso de arrendamiento. En caso de que el monto supere la garantía, o no exista dicha garantía, se realiza un cobro por los daños y perjuicios que existan dentro del inmueble. Este cobro puede ser aceptado o no por el arrendatario, si este último acepta el cobro, se realizan los pagos correspondientes, se hace abandono de la propiedad y se termina el proceso. Si por el contrario, no existe una resolución que satisfaga a ambas partes, se lleva a juicio en los Juzgados Civiles (Demanda Por Deuda De Arriendo - Ley Fácil - Biblioteca Del Congreso Nacional De Chile, 2008), aquí depende del monto del arrendamiento si se necesitan abogados o no, si el monto es superior a 4 UTM, entonces es necesario la presencia de profesionales jurídicos en el juicio, de lo contrario simplemente las partes en cuestión

presentan los hechos y el juzgado realiza la resolución final, según la decisión del juzgado, se finaliza el proceso de arrendamiento.

1.2.2 Proceso de pago de cuentas

Para el proceso de pago de cuentas, se deben tener claro los participantes del proceso:

- Propietario o administrador: persona u organización, que es dueña de la propiedad y debe cumplir con los pagos de los gastos que esta genera, este participante puede ser reemplazado por un tercero que por medio de un acuerdo con el propietario es responsable de mantener la propiedad con sus cuentas al día, a cambio de un pago monetario.
- Proveedor de Servicio: es la empresa que se encarga de entregar un servicio a una propiedad específica.

En la actualidad, existen dos tipos de cuentas, la que está adherida al propietario de la propiedad, como lo puede ser la cuenta de servicios de internet y televisión, y las cuentas que están conectadas directamente a la propiedad como lo son la luz o el agua. Sin embargo, estos tipos de cuentas se pueden juntar y convertir en una sola en función al acuerdo que llegue el administrador de la propiedad con el propietario. Un ejemplo de ello es la cuenta de internet, la cual se asocia a una persona, que en este caso sería al propietario, sin embargo el servicio se le entrega a la propiedad que está siendo manejada por el administrador, entonces si el acuerdo entre las partes explicita que el administrador se hará cargo de ese gasto en particular, entonces el administrador lo agregara a la lista de gastos asociados a la propiedad, con el objetivo de mantener todos los pagos de los servicios de la propiedad al día.

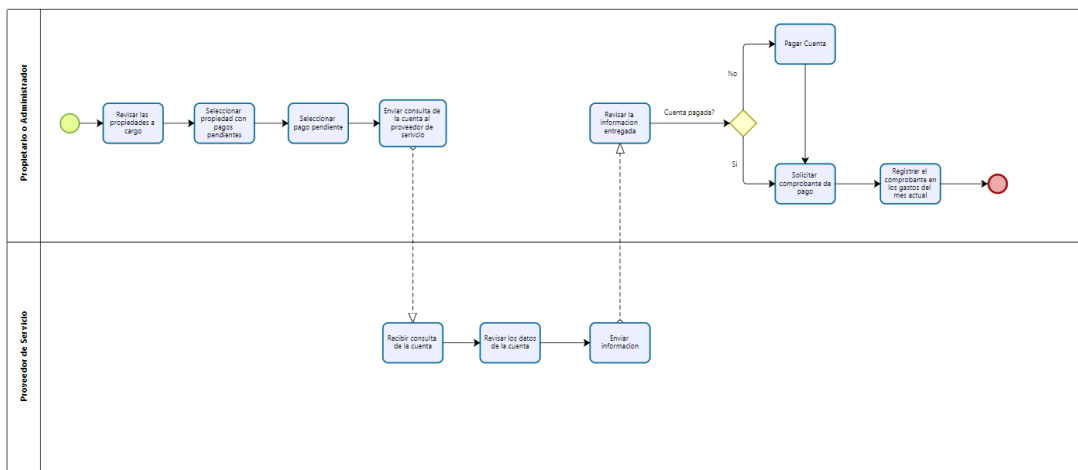


Figura 3. Diagrama BPMN que refleja el proceso de pago de cuentas.

Fuente: elaboración propia

Como se puede observar en la figura 3, el administrador o propietario se encarga de revisar las propiedades a su cargo, selecciona una y observa que cuentas no han sido

pagadas o registradas, seleccionada la cuenta a pagar envía una solicitud al proveedor para verificar que el pago aún no se ha realizado, en el caso de que ya se encuentre pagado solicita el comprobante y lo registra. Sin embargo, si el pago aún no está realizado, procede a generar el pago, obtener el comprobante de este y lo agrega a la lista de gastos del mes actual.

1.2.3 Proceso mantenimiento de propiedades

Para el proceso de mantención de propiedades, se debe definir qué actores son los que participan en el proceso. Algunos de estos son:

- Propietario o administrador: persona u organización, que se encarga de la propiedad revisando que se encuentre en buen estado para su uso.
- Maestro: Persona u organización que se encarga de realizar labores de maestranza en propiedades varias.
- Arrendatario: Persona que se encuentra arrendando la propiedad durante una incidencia o mantención programada.

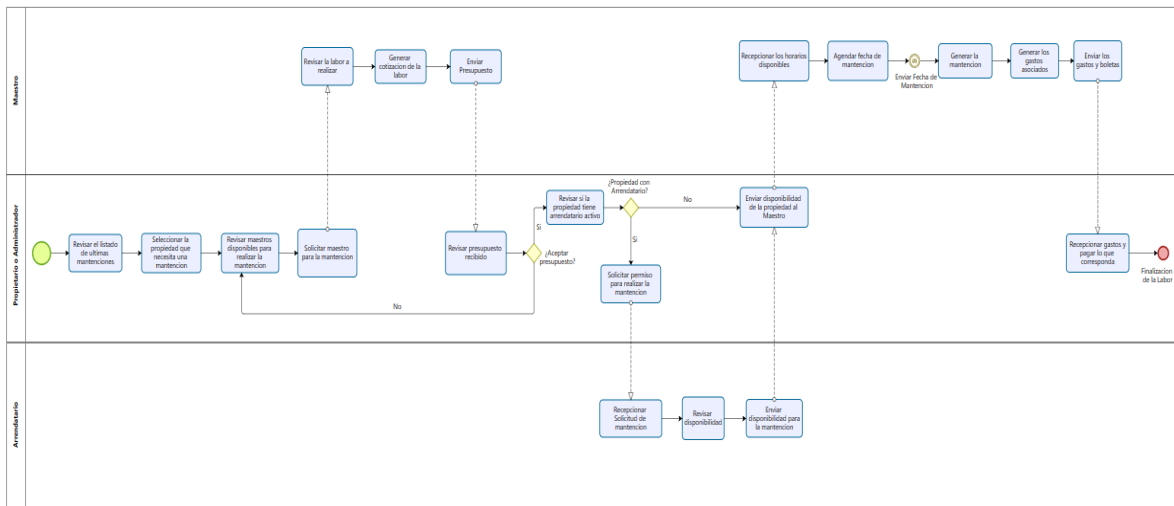


Figura 4. Diagrama BPMN que refleja el proceso de mantenimiento de propiedades.

Fuente: Elaboración Propia

Como se observa en la figura 4, el administrador o propietario se encarga de revisar el listado de mantenciones realizadas, selecciona una de las propiedades que necesita una mantención y revisa a los maestros que tiene disponible para generar la labor, una vez selecciona el maestro, solicita al maestro que le genera una cotización por la labor a realizar, el maestro se la envía para que el administrador la revise y la valide, en caso de rechazarla, el administrador procede a buscar otro maestro para la labor siguiendo el proceso antes mencionado, en caso de que se acepten las condiciones de la mantención se continua con la revisión de la disponibilidad de la propiedad, ya que esta puede estar con un arriendo activo, lo que provoca que se tenga que coordinar con el arrendatario la fecha para que el maestro pueda ir a la propiedad a realizar la mantención, en caso contrario simplemente se coordina con el maestro, una vez enviada la disponibilidad al

maestro, este último selecciona la fecha para realizar la mantención, informa a las partes participantes, genera la labor, informa al administrador sobre los gastos reales del trabajo, el administrador verifica el trabajo, registra los gastos generados y da por finalizada la mantención de la propiedad.

En el caso de que se necesite un arreglo por un siniestro en la propiedad, se realiza el mismo procedimiento.

1.2.4 Proceso de registro y aviso de pago de arriendo

Para comprender el proceso del registro y aviso de pago de arriendo, se deben considerar a los actores que participan en el proceso. Estos son los siguientes:

- Propietario o administrador: persona u organización, que por medio de un acuerdo con el arrendatario se encarga de recibir el pago por concepto de arriendo de la propiedad.
- Arrendatario: persona que hace uso de la propiedad durante un tiempo determinado a cambio de un pago especificado en el contrato.

Para este proceso, se considera que el arrendatario tiene que cumplir con lo estipulado en el contrato de arriendo, el cual indica que tiene plazo hasta cierto día del mes para pagar el arriendo del mes que transcurre, por ejemplo si el contrato indica que el arrendatario debe pagar el arriendo los días 27 de cada mes, entonces el arrendatario tiene la obligación de cumplir con su deber, sin embargo, puede existir la posibilidad de que el administrador envíe un aviso al arrendatario activo para recordarle que el plazo del arriendo esta por vencer y que pagando antes de la fecha límite, evita posibles multas que en caso de existir deben estar descritas en el contrato de arriendo.

Considerando lo anterior, existen dos procesos los cuales se detallan a continuación. El proceso de pago de arriendo oportuno, que trata de que el arrendatario cumple con su obligación de pagar el arriendo antes de la fecha establecida, enviando su comprobante de pago al administrador para que este pueda guardarlo y/o imprimirlo, con el objetivo de respaldar la información enviada por el arrendatario. El proceso se representa en la siguiente figura.

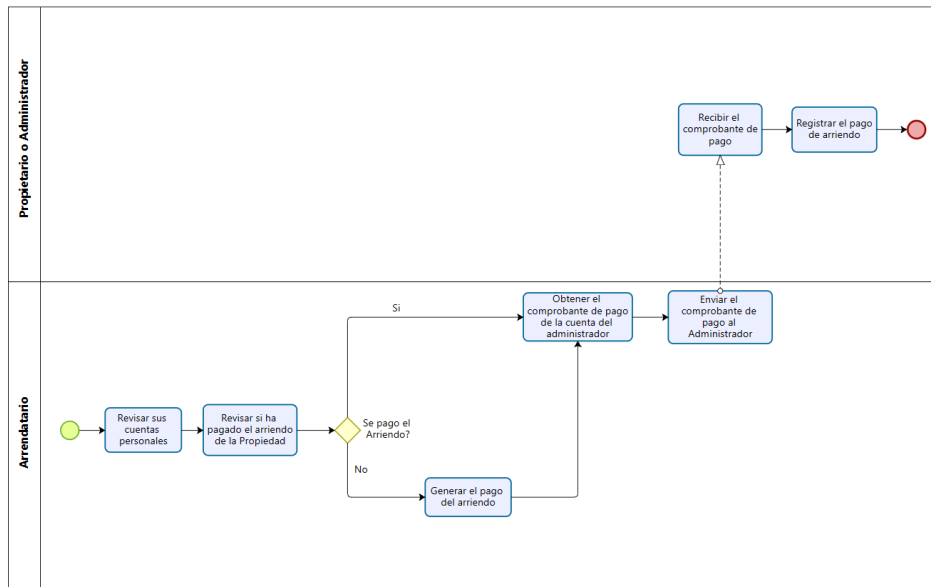


Figura 5. Diagrama BPMN que refleja el proceso de pago de arriendo.
Fuente: elaboración propia

En cambio, está el otro caso que consiste en que el administrador revisa los pagos de arriendo de las propiedades que administra, y observa que existe una propiedad que aún no envía su comprobante de pago, y, en consecuencia, no se ha registrado el pago del arriendo. Este caso se detalla en la siguiente figura.

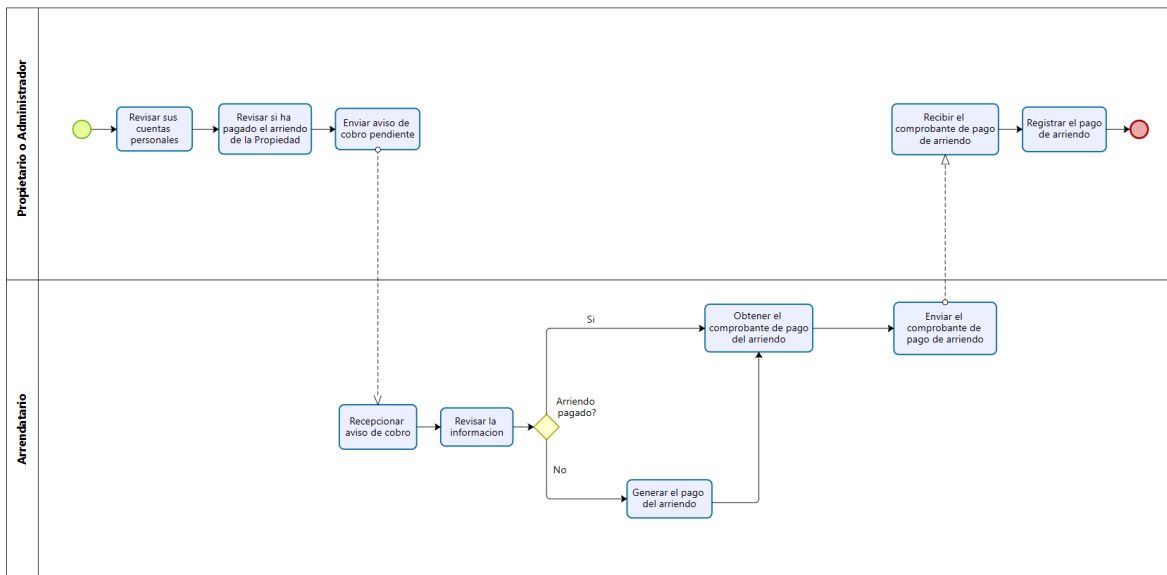


Figura 6. Diagrama BPMN que refleja el proceso de aviso de no pago de arriendo y el pago de este.
Fuente: elaboración propia

En la Figura 6, se puede observar que el administrador tiene que revisar su cartera de propiedades y revisar qué pagos no se han realizado de forma oportuna, se selecciona la propiedad de interés y se le envía un aviso al arrendatario para que realice el pago correspondiente o envíe el comprobante en caso de haber realizado el pago y se le haya olvidado enviarlo, en caso de que el arrendatario no haya realizado el pago, se procede a generar el pago y enviar el comprobante al administrador de la propiedad, este último lo recibe, lo registra y termina el proceso de pago de arriendo.

1.2.5 Proceso de generación del estado de cuenta de una propiedad

Para comprender el proceso de generación del estado de cuenta de una propiedad o conjunto de propiedades, primero se debe definir qué actores son los que participan en el proceso.

- Propietario o administrador: persona u organización, que se encarga de realizar el estado de cuenta de las propiedades, considerando los ingresos y egresos que genera la propiedad, en caso de que el actor sea un administrador, este debe enviar el estado de cuenta a los propietarios de la propiedad.

Para este proceso, el administrador o propietario se encarga de recopilar toda la información generada de los procesos anteriores, los ingresos generados por el pago del arriendo y otras fuentes de ingreso, y los egresos de la propiedad, generados por los pagos de cuenta adheridos a la propiedad, el mantenimiento y los arreglos de la propiedad. Una vez recopilada esta información se genera un consolidado de gastos e ingresos, y se obtiene el estado de cuenta de la propiedad. Todo este proceso conlleva una gran inversión de tiempo debido a que se deben recopilar datos de diversas fuentes de información, y además, se debe elaborar un informe que contemple cada uno de los gastos e ingresos que se generan mes a mes.

1.3 Problemáticas generales

Los procesos que se explican anteriormente permiten dilucidar diferentes complicaciones que aparecen en la vida de los administradores de bienes raíces y las personas que quieren comenzar en el mundo de los negocios inmobiliarios, esto sin considerar que las leyes chilenas exigen una serie de datos indispensables para que el posible contrato de arrendamiento pueda ser legalizado frente a notario, estos antecedentes son:

- Propiedad
 - Ubicación exacta
 - Descripción del Inmueble
 - Especificar las condiciones reales de la propiedad
 - Fotografiar a la hora de la entrega del inmueble
- Arrendador
 - Datos Generales
 - Nombre y Apellido

- Dirección
- Tipo de Persona (natural o jurídica)
- Certificación de dominio de la propiedad
 - Escrituras del inmueble
 - Certificado de dominio que entrega el Conservador de Bienes Raíces
- Arrendatario
 - Datos Generales: Datos mínimos sobre la persona que hará uso de la propiedad, estos antecedentes son:
 - Nombre y Apellido
 - Dirección
 - Tipo de persona (natural o jurídica).
 - Documentos Generales: Documentos que permiten saber la situación financiera del arrendatario
 - RUT
 - Certificado de DICOM.
 - Contrato de trabajo Indefinido
 - Últimas 3 Liquidaciones de sueldo
 - Últimas 3 Cotizaciones AFP
 - Declaraciones de Renta
 - Datos del aval
- Contrato: acuerdo escrito, en el que dos o más partes se comprometen recíprocamente a respetar y cumplir una serie de condiciones, tales como:
 - Monto para pagar por renta
 - Tipo de pago:
 - Temporalidad
 - Forma de Pago
 - Existencia de Reajuste del monto.
 - Plazo del contrato
 - Responsabilidades de ambas partes en el mantenimiento del inmueble
 - Cláusulas de término de contrato en caso de que una de las partes incumpla lo pactado

Debido a la gran cantidad de datos y papeles que se necesitan para poder realizar el trámite de arrendamiento, es que según el banco Central (*El mercado de arriendo residencial*, n.d.), muchos de los arriendos se realizan de forma privada, sin recurrir a contratos ni a intermediarios que fiscalicen el acuerdo entre las partes, provocando un aumento dentro de las demandas por incumplimiento de acuerdos de arriendo, ya que según la corte suprema, cada año se presentan más de 15 mil juicios por arrendamiento de los cuales solo 1 de cada 4 son realmente finalizadas, además de que el miedo dentro de los pequeños y grande inversionistas inmobiliarios ha ido en aumento, debido a que se cree que las leyes de Chile protegen más a los individuos morosos que a los propios arrendadores, lo cual genera una pérdida dentro de sus propias inversiones (DF MAS, 2021), entonces teniendo en cuenta los datos antes mencionados, es que se muestra que es un problema de suma urgencia que no debe ser tomado a la ligera.

Lo mencionado anteriormente deja en evidencia una problemática importante en el área inmobiliaria, la gran cantidad de documentos físicos que se tienen que recopilar, provocan una lentitud a la hora de la entrega e intercambio de información entre todas las partes, esto es mostrado en un estudio realizado por Developing People Globally, el cual menciona que el 57% de los trabajadores de oficina pierde al menos 1 hora al día buscando documentos (Melo, 2020), este tiempo solo considera la búsqueda de estos documentos, excluyendo tiempos de impresión y de traslado para realizar el intercambio de información y la firma de un contrato, este estudio considera además del tiempo de búsqueda, un costo económico para la empresa o persona, que se podría ahorrar, evitando el gasto en tinta, papel, traslado y personal, que a través de la digitalización se podría reducir o hasta eliminar.

Por otro lado, la seguridad de los datos y documentos es fundamental para lograr un buen manejo de la propiedad, además permite ante cualquier problema que surja, tener un respaldo que ayude a solucionar las discrepancia entre ambas partes, sin embargo, al ser muchos documentos y que el formato de los papeles sea físico, es que puedan existir problemas con ellos, debido a que estos papeles puedan ser olvidados o hasta robados, lo cual genera una pérdida de tiempo y de ingresos que podría haber sido evitada con una herramienta que digitalice cada uno de los papeles y que además podrían ser accedidos desde cualquier lugar con acceso a internet.

Teniendo en cuenta el proceso de arrendamiento, existe una variedad de contratos de esta índole, los cuales cambian según las condiciones que se estipulan dentro del mismo, actualmente en Chile hay tres tipos de contrato de arrendamiento:

1. Contrato a plazo fijo sin opción a renovación
2. Contrato a plazo fijo renovable después de 12 meses
3. Contratos mensuales o por tiempo indefinido

Debido a los tipos de arrendamientos es que en general, los arrendatarios cambian constantemente de propiedad, lo cual, mantener un historial de quienes utilizaron la propiedad en cierto periodo es un problema complicado de abarcar por los administradores, y es mucho mayor si no hay un control estricto de la propiedad, además si el administrador está a cargo de más propiedades, el problema se vuelve más complejo y difícil de abordar. Por otro lado, debido a la nula comunicación con el arrendatario luego del término del contrato, no es posible tener una cartera de clientes consistente, ya que los datos de contacto de los antiguos arrendatarios cambian y no son estáticos, entonces si un arrendatario cambio su teléfono, el administrador queda sin posibilidad de contactarlo en un futuro, lo cual genera la pérdida de un posible negocio.

Otra problemática que se puede dilucidar del dinamismo de arrendatarios es el control del estado de una propiedad, no se sabe con exactitud en qué condiciones está el inmueble antes de entregarlo ni después de ser recuperado. Es una responsabilidad de los administradores mantener una visualización actualizada y real del inmueble, lo cual en caso de no existir puede traer problemas entre el arrendatario y el arrendador, ya que no

muestra transparencia a la hora del acuerdo ambos, y además no muestra la realidad de la propiedad, provocando discusiones entre las partes, respecto al estado de la propiedad antes y después de entregarla, en el peor de los casos, si no existe un acuerdo entre ellos y no hay un respaldo genuino que verifique que realmente la propiedad estaba en las condiciones que fue cedida, puede generar problemas judiciales como los que se explicitan en el proceso de arrendamiento.

1.4 Problemáticas específicas del cliente.

El software EasyKey, tiene por objetivo solucionar diferentes problemas que aquejan al rubro inmobiliario, como también a cumplir ciertas dificultades que afectan al cliente Inversiones Arrecife, empresa que apoya esta memoria, ellos buscan solucionar las problemáticas que enfrentan día tras día en el rubro inmobiliario, entre ellas están:

- **Registros totalmente físicos:** todos los comprobantes, contratos y documentos relevantes para la generación de un arriendo están en formato físico sin respaldo alguno.
- **No existe un canal centralizado de información:** no existe un registro digital ni centralizado de la información de los arriendos, lo que genera una dependencia del personal a cargo, ya que las personas responsables de la propiedad son las únicas que manejan esa información, lo cual provoca, que en caso de que el personal a cargo sufra algún inconveniente, nadie se pueda hacer cargo de la propiedad ni tenga la información disponible para tomar la responsabilidad del bien raíz.
- **Nulo registro de las propiedades a su responsabilidad:** actualmente no tienen un registro de las propiedades que tienen activas, ni el estado en el que se encuentran, además carecen de un historial de reparaciones, lo cual no permite una administración adecuada de las propiedades.
- **Nulo registro de gastos e ingresos de las propiedades:** existe una gran cantidad de comprobantes de pago y gastos sin registrar, lo cual no permite que las liquidaciones que se envían a los inversionistas sea una representación fidedigna de la realidad de cada propiedad, lo cual genera descuadres en el estado de cuenta de cada propiedad y en consecuencia de cada propietario.
- **Generación de los estados de cuenta de los propietarios de forma manual:** en la actualidad, mes a mes los trabajadores de la empresa Arrecife realizan los estados de cuenta de cada propiedad y un consolidado para los propietarios, estos consolidados toman bastante tiempo, lo cual es un gasto para la empresa en cuestión.
- **No existe un registro real de maestros:** no hay un registro concreto de los maestros que trabajan con la empresa ni un detalle de las especialidades de cada uno.
- **No se generan órdenes de trabajo para los maestros:** actualmente los maestros funcionan en base a llamadas y boletas, por lo cual no hay un registro de las labores ni de los gastos realizados por los profesionales.
- **No existe un historial de reparaciones:** no mantienen un registro de las labores realizadas por los maestros, por lo cual, no existe un historial que permita saber en

qué momento las propiedades pueden necesitar alguna mantención o reparación que sin ella puede resultar en una complicación mayor.

1.5 Mercado actual.

Debido a las problemáticas antes mencionadas se debe realizar un estudio respecto a las soluciones que existen en la actualidad para generar un software que solucione las problemáticas antes mencionadas y tenga una propuesta de valor distinta a las demás.

1.5.1 Mercado Chileno

En Chile, existen tres tipos de software que se enfocan en el arriendo inmobiliario. Estos son los siguientes:

- **Leasity:** aplicación web, que se enfoca en el arriendo fue desarrollado por la empresa del mismo nombre, se enfoca principalmente en los gastos e ingresos de la propiedad y en captar nuevos arrendatarios.
- **FairRent:** aplicación de escritorio, desarrollada por Fairware y se enfoca en la administración de propiedades.
- **BReal:** aplicación web, desarrollada por BReal Estate, se enfoca en la agilidad y el control de procesos del área inmobiliaria.

A continuación, se muestra un cuadro comparativo con los requerimientos que debe cumplir el software y sus respectivos precios.

Requerimientos	Leasity	FairRent	BReal	EasyKey
Respaldo de Gastos e ingresos	SI	SI	SI	SI
Respaldo de documentos relevantes	SI	SI	SI	SI
Respaldo de los Arreglos de la Propiedad	NO	SI	SI	SI
Registro de Maestros de confianza	NO	NO	NO	SI
Respaldo del	NO	NO	NO	SI

estado al entregar la propiedad				
Automatización de la generación de los estados de cuenta.	SI	SI	SI	SI
Suscripción o Pago Único	Suscripción Mensual	Suscripción Mensual	Suscripción Mensual	Suscripción Mensual
Precio	Desde 1 UF - 20 Propiedades	Sin información	Sin información	2 UF - Sin restricción de Propiedades

Tabla 1. Comparativa entre los tipos de software que están presentes en el mercado chileno y EasyKey.

Fuente: elaboración propia

1.5.2 Mercado internacional.

Actualmente existen varios tipos de software de administración de propiedades en todo el mundo, los que más se destacan son los siguientes:

- **Buildium** Aplicación web y móvil, que está enfocada en la administración de bienes raíces, desarrollada en Estados Unidos por la empresa Buildium, realiza un manejo de propiedades, entrega avisos y realiza un manejo de la contabilidad de las propiedades agregadas al sistema.
- **Rentger:** Aplicación web, software focalizado a la gestión de arriendos de propiedades, realizando la contabilidad de los gastos e ingresos de la propiedad, revisando a posibles postulantes y realizando la gestión de los contratos a firmar, es un desarrollo generado en España por una empresa del grupo Idealista.
- **AppFolio:** Aplicación móvil, que realiza la gestión de los propietarios y residentes de las propiedades añadidas, considerando la contabilidad general de la propiedad y fue desarrollada en Estados Unidos por AppFolio inc.

Las aplicaciones antes mencionadas, pueden solucionar las problemáticas generales y específicas del cliente, sin embargo, se deben considerar varios factores que pueden dificultar la implementación de una solución en un país extranjero:

- **Barreras lingüísticas.** Dos de las tres aplicaciones mencionadas son de USA, que tiene como idioma predominante el inglés, lo cual puede generar una dificultad

para la implementación de la solución en un país que no tiene el mismo idioma, como también la comunicación con el soporte técnico puede generar dificultades y provocar un gran trauma en el cliente final.

- **Diferencias entre las leyes de cada país.** Cada país tiene leyes distintas que cumplir y además estas leyes pueden ir variando a través del tiempo, lo que puede generar que en un inicio la solución satisfaga la mayoría de los requerimientos solicitados, sin embargo una modificación dentro de las leyes del país, podría generar que se soliciten cambios en el software, solicitar estos cambios a una entidad que se encuentra en otro país, puede ser dificultoso, hasta incluso generar cobros extras por los desarrollos a realizar.
- **La conversión de las monedas.** La conversión de la moneda es un tema relevante en el ámbito inmobiliario, ya que se manejan varias cifras y las conversiones entre monedas puede generar problemas a la hora de generar arriendos y pagos en las propiedades.

Por lo anterior, es que se busca generar una solución óptima que abarque las problemáticas y los desafíos que se han encontrado en esta investigación.

1.6 Objetivos

1.6.1 Objetivo general

- Desarrollar una plataforma de fácil uso que permita a los administradores de bienes raíces manejar diversas propiedades, manteniendo un estado de caja controlado y ordenado.

1.6.2 Objetivos específicos

- Proponer, por medio de una plataforma web, una metodología de trabajo que le permita al administrador tener un estado de cuenta actualizado e informe a los propietarios sobre sus propiedades.
- Aplicar metodologías y tecnologías de desarrollo de software que permitan la construcción de la plataforma de acuerdo con el diseño previamente establecido.
- Generar un sistema que centralice toda la información relevante para la administración de bienes raíces, disminuyendo la dependencia del personal humano para conocer el estado de cada propiedad.
- Realizar pruebas de usabilidad que permitan garantizar que la plataforma sea intuitiva y fácil de utilizar para diferentes usuarios.

1.7 Alcances y desarrollo de la memoria

La memoria se enfoca principalmente en el ámbito del arriendo de una propiedad, la transformación digital de todo el proceso y la construcción de la plataforma para la solución de las problemáticas detalladas anteriormente, sin embargo, el desarrollo seguirá su curso luego del término de la memoria, para lograr el módulo de ventas y así lograr la solución final del software, cumpliendo los objetivos marcados anteriormente.

Esta memoria es apoyada por la empresa de desarrollo CleverSystem SPA, una empresa que inició en el año 2019 y se ha abierto al mercado enfocándose en desarrollar soluciones que permitan que distintas empresas, pymes o personas puedan unirse a la tan famosa transformación digital, es por lo anterior y realizando un análisis del mercado se vio una oportunidad importante de insertarse dentro de un mercado tan afectado por la pandemia y generar un cambio importante en una de las áreas menos abarcadas por la digitalización.

CAPÍTULO 2: MARCO CONCEPTUAL.

En el capítulo anterior se muestran las problemáticas y los procesos de la administración de propiedades. En este contexto, surge la necesidad de utilizar la transformación digital como un enfoque estratégico que permita superar los desafíos identificados y optimizar la eficiencia en esta área. La integración de tecnologías avanzadas y soluciones digitales permitirá una gestión más ágil, precisa y colaborativa de las propiedades, redefiniendo fundamentalmente la forma en que se llevan a cabo las operaciones y se interactúa con los diferentes aspectos del proceso de administración. Para ello se requiere comprender lo que es la transformación digital, los conceptos de desarrollo de software y la experiencia de usuario

2.1 Transformación digital

La transformación digital es el proceso mediante el cual una organización implementa tecnologías digitales en sus métodos tradicionales, generando cambios desde las operaciones de la empresa hasta la cultura laboral de la misma, todos estos cambios tienen el objetivo de mejorar y optimizar las labores de la organización.

Esta transformación no solo implica la implementación de nuevas herramientas, sino que muchas veces genera una redefinición de los flujos de trabajos de la empresa, modificando la forma en que se hacían los procesos en un inicio, lo que provoca un cambio cultural en la forma de trabajar de cada integrante de la empresa.

En este documento se buscan transformar cinco procesos que se expusieron en el capítulo anterior.

2.1.1 El proceso de Arriendo de una Propiedad

Para el proceso de arriendo se busca recopilar la información de todos los arriendos dentro de un sistema centralizado y que estos antecedentes queden disponible para todos los integrantes de la empresa, ya que como se mencionó en el capítulo anterior, la búsqueda de documentación física genera una gran pérdida de tiempo dentro de una empresa.

2.1.2 Proceso de pago de cuentas

En el proceso de pagos de cuentas se busca centralizar la obtención de información sobre la propiedad, esto se debe a que cada cuenta, sea de luz, de agua, u otras, tiene un identificador único y diferente, lo que dificulta englobar las cuentas por propiedad, entonces lo que se busca es disponibilizar las cuentas de cada propiedad y verificar que estas se encuentren al día.

2.1.3 El proceso mantenimiento de propiedades

Al igual que el proceso de arriendo, toda la información respecto a las propiedades se encuentra dispersa y sin ningún respaldo digital, por lo cual en este caso se busca registrar los gastos creados por la mantención y los siniestros que generan las propiedades.

2.1.4 Proceso de registro y aviso de pago de arriendo

Para este proceso se busca centralizar la información de los pagos de arriendo de cada propiedad y automatizar la funcionalidad de aviso de pago de arriendo, ya que actualmente no existe un método automatizado que ayude al administrador a notificar a sus arrendatarios sobre el estado de su arriendo en el mes transcurrido.

2.1.5 Proceso de generación del estado de cuenta de una propiedad

Para el proceso de estado de cuenta, se busca automatizar la generación del estado de cuenta de una propiedad, ya que actualmente se registran todos los movimientos de ingresos y egresos de una propiedad en particular, y el informe se genera de forma manual mes a mes, lo que toma bastante tiempo debido a que las fuentes de datos no están centralizadas y el traspaso de información entre los colaboradores no es fiable, por lo cual se busca que con los datos recopilados de los procesos anteriores permitan generar los estados de cuentas de cada propiedad.

Para la adopción de nuevas tecnologías en los procesos mencionados se va a generar un software que busca la transformación digital en el área de la administración inmobiliaria, para ello es esencial entender las metodologías que guían el desarrollo de soluciones digitales.

2.2 Metodologías de desarrollo de software

Las metodologías de desarrollo son un conjunto de técnicas y métodos que son utilizadas para diseñar un software, actualmente existen 2 sistemas de metodologías, los métodos tradicionales y los métodos ágiles.

2.2.1 Metodologías tradicionales

Las metodologías tradicionales se enfocan en seguir un proceso secuencial riguroso que pasa por diferentes etapas, generando el ciclo de vida del software, lo anterior está conformado por las siguientes fases:

- Inicio
- Planificación
- Ejecución
- Seguimiento
- Cierre

En general, estas metodologías se enfocan principalmente en cumplir el ciclo del proyecto, entregando mayor importancia a los costos, el alcance y el tiempo del desarrollo de la aplicación, dejando los requisitos del cliente en un segundo plano y sin posibilidad de cambios. Entre los métodos tradicionales más utilizados están:

- Metodología espiral
- Metodología cascada
- Metodología de prototipo

Las metodologías tradicionales tienen ventajas como desventajas, a continuación se muestran algunas de ellas.

- **Ventajas**

- **Planificación detallada:** las metodologías tradicionales se centran en una planificación exhaustiva, lo que puede ayudar a tener una visión clara del proyecto desde un inicio.
- **Control de riesgos:** al seguir un enfoque secuencial, se pueden identificar y mitigar los riesgos antes de avanzar a la siguiente fase.
- **Documentación completa:** las metodologías tradicionales se enfocan en realizar una documentación detallada, lo que facilita el seguimiento y la comprensión del proyecto.

- **Desventajas**

- **Poca flexibilidad:** los cambios o nuevos requisitos pueden ser difíciles de incorporar una vez que se ha completado una fase, lo que puede llevar a retrasos en el proyecto.
- **Poca adaptabilidad al cambio:** los proyectos pueden enfrentar dificultades si hay cambios significativos en los requisitos o el alcance del proyecto durante su desarrollo.
- **Riesgo de falta de retroalimentación temprana:** al seguir un enfoque secuencial, la retroalimentación de los usuarios finales o clientes puede no ocurrir hasta etapas posteriores, lo que podría generar problemas de adaptación

2.2.2 Metodologías ágiles

Por otro lado están las metodologías ágiles, las cuales se enfocan en el trabajo en equipo, la colaboración y la flexibilidad de los clientes, generando procesos iterativos centrados en los comentarios de los clientes y las versiones continuas que genera cada iteración del proyecto. Esto permite que el trabajo sea flexible y adaptable, enfocando el desarrollo en la solución de los requerimientos del cliente. Algunas metodologías ágiles son:

- Metodología Scrum

- Programación Extrema (XP)
- Metodología Kanban

Cada metodología tiene su propio proceso por ejemplo, la metodología Scrum, se enfoca en la división y la autonomía del equipo de desarrollo, una parte del equipo se encarga en el desarrollo del producto y el otro es responsable del progreso y la planificación del mismo, en cambio, la metodología Kanban, apunta a generar cargas de trabajo que se deben respetar hasta que finalice, ya que las cargas son inmutables, luego de que termine la lista de tareas actual, se pueden armar una nueva carga para realizar cambios en la aplicación, debido a la gran diferencia entre ambas corrientes de desarrollo, es que no existe un proceso estandarizado para las estrategias ágiles.

Las metodologías ágiles al igual que las tradicionales, tienen ventajas como desventajas, a continuación, se muestran algunas de ellas.

- **Ventajas**

- **Flexibilidad y adaptabilidad:** las metodologías ágiles se centran en responder rápidamente a los cambios, lo que permite ajustar los requisitos y prioridades a medida que se avanza en el proyecto. Esto brinda una mayor flexibilidad y adaptabilidad a las necesidades del cliente.
- **Entrega temprana de valor:** las metodologías ágiles se basan en la entrega de incrementos funcionales de software de manera continua. Esto permite entregar valor a los usuarios finales en etapas tempranas del proyecto, lo que puede generar retroalimentación y validación temprana.
- **Mayor participación del cliente:** las metodologías ágiles promueven una mayor colaboración y participación del cliente a lo largo del proyecto. Los clientes o usuarios finales están más involucrados en la toma de decisiones, brindando una retroalimentación continua y contribuyendo a la satisfacción de sus necesidades.
- **Mayor enfoque en la calidad:** las metodologías ágiles promueven prácticas de desarrollo de software como pruebas continuas, integración continua y revisión de código. Esto permite una mayor atención a la calidad del software entregado.
- **Mejora de la comunicación y el trabajo en equipo:** las metodologías ágiles fomentan la comunicación frecuente y la colaboración entre los miembros del equipo. Esto ayuda a crear un entorno de trabajo colaborativo, que promueve la sinergia y la resolución eficiente de problemas.

- **Desventajas:**

- **Requiere una buena comunicación y colaboración entre los participantes:** el enfoque ágil requiere una comunicación efectiva y constante entre los miembros del equipo y el cliente final. Esto puede ser un desafío en equipos con una comunicación deficiente o clientes con poca disposición a reunirse.

- **Riesgo de desviarse del alcance:** la flexibilidad y la adaptabilidad de las metodologías ágiles pueden llevar a cambios frecuentes en el alcance del proyecto. Esto requiere un equilibrio cuidadoso para evitar desviaciones significativas y asegurar que los objetivos del proyecto se cumplan.
- **Mayor dependencia del conocimiento del equipo:** las metodologías ágiles requieren que los miembros del equipo sean competentes y tengan un buen conocimiento técnico. La falta de experiencia o conocimientos adecuados puede afectar negativamente a la implementación de una metodología ágil.
- **Requiere una buena gestión del tiempo y los recursos:** las metodologías ágiles requieren una planificación y gestión adecuada del tiempo y los recursos. El cumplimiento de los plazos y la asignación eficiente de los recursos son fundamentales para el éxito de un proyecto ágil.

Cabe destacar que todos los métodos antes mencionados tienen características que les brindan ventajas y desventajas al momento de realizar un desarrollo, lo que las diferencia entre ellas, esto les permite ajustarse al ecosistema del proyecto de forma distinta, por lo cual es fundamental que el equipo o el jefe de proyecto decida qué metodología es la mejor para afrontar el desarrollo, ya que el método a utilizar será fundamental a la hora de abordar los requerimientos del cliente, estimar los tiempos del desarrollo, optimizar el uso de los recursos disponibles, la estimación de los costos, entre otros.

2.2.3 Metodología programación extrema (XP)

Para el desarrollo de la solución, se ha optado por utilizar la metodología de programación extrema (XP). Esta metodología se caracteriza por fomentar un ambiente de colaboración estrecho entre el equipo de desarrollo y el cliente, con el objetivo de lograr una mayor comprensión de los requisitos y expectativas del proyecto. La empresa de desarrollo debe estar preparada para adaptarse a posibles cambios que surjan durante las interacciones con el cliente, ya que la flexibilidad es una de las premisas fundamentales de XP. Además, la metodología permite realizar entregas rápidas y frecuentes, lo que involucra activamente al cliente en el proceso de desarrollo. Esta participación activa le permite al cliente revisar y proporcionar retroalimentación sobre las entregas, lo que ayuda a reducir posibles problemas o cambios significativos en módulos ya finalizados y difíciles de modificar. De esta manera, la metodología XP busca minimizar los contratiempos y asegurar la entrega de un código desarrollado de manera eficiente y en línea con las expectativas del cliente.

La metodología XP es una metodología ágil de desarrollo de software que consta de 5 fases principales, las cuales se presentan en la siguiente imagen:

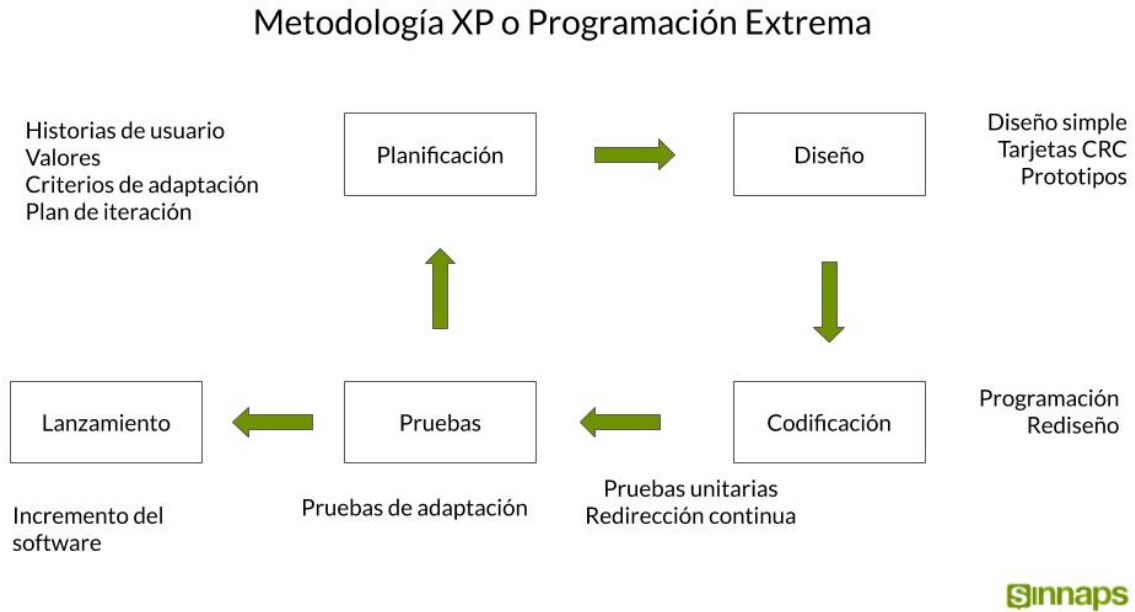


Figura 7. Elaborada por Sinnaps, Diagrama de flujo de la metodología XP
Fuente: <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp>

1. Fase de planificación

En la fase de planificación del proyecto, se lleva a cabo la identificación y definición de los requisitos. Para lograr esto, se realizan reuniones con el cliente, donde se llevan a cabo entrevistas y discusiones para recopilar la información necesaria y comprender las problemáticas que el cliente enfrenta. Durante esta fase, se generan las historias de usuario, que son explicaciones generales e informales de funciones específicas del software a desarrollar, redactadas desde la perspectiva del usuario final, estas funcionalidades son explicadas por medio de la siguiente estructura:

- **Título:** título que describe la funcionalidad deseada.
- **Descripción:** detalles adicionales sobre la historia, incluyendo el contexto, las necesidades del usuario y los objetivos a lograr.
- **Criterios de aceptación:** condiciones específicas que deben cumplirse para considerar la historia como completada y funcional.

2. Fase de diseño

En la fase de diseño, se trabaja con la información recopilada durante la fase de planificación. Se revisan las historias de usuario creadas y se comienza a generar el primer prototipo estable del sistema. Se genera una planificación y se establecen fechas para las reuniones con el cliente, donde se presentarán los

avances realizados en el sistema, en general estas reuniones son cada 1 o 2 semanas.

3. Fase de codificación

En esta etapa, el sistema comienza a crecer y a adquirir robustez en sus funcionalidades. Se mejoran aspectos visuales, se cumple con los criterios de aceptación de las historias de usuario y se realizan entregas para que el cliente pueda interactuar con el sistema.

4. Fase pruebas

Esta fase inicia con las primeras entregas al cliente, con el objetivo de que pueda revisar los avances generados y proporcionar retroalimentación oportuna al equipo de desarrollo. En esta etapa, el sistema ya adquiere forma y el cliente puede comenzar a utilizar algunas funciones de la aplicación, lo cual le permite familiarizarse con el software y comprender el flujo del sistema.

5. Fase de lanzamiento

En esta fase, el sistema ya cuenta con todas las funcionalidades solicitadas por el cliente y cumple con sus expectativas. En este punto, se consideran los acuerdos comerciales entre la empresa de desarrollo y el cliente para determinar cómo se abordarán posibles errores (bugs) que se presenten en el sistema.

2.3 Arquitectura de software.

Continuando, se deben considerar al igual que con las metodologías, que arquitecturas de software se utilizan en la actualidad, la arquitectura de software, es un concepto que surge ya en los años 60 y se refiere a una planificación basada en modelos, patrones y abstracciones teóricas, a la hora de realizar una pieza de software de cierta complejidad y como paso previo a cualquier implementación. (Huet, P. 2022, agosto 24. Arquitectura de software: Qué es y qué tipos existen). Principalmente la arquitectura de software se enfoca en la forma en que se realiza la comunicación entre el cliente y la máquina, de qué forma se realiza el envío de los datos al servidor, como se realizan los cálculos por detrás y todo lo que tiene que ver entre la interfaz gráfica y los motores de bases de datos, algunos de los patrones de arquitectura más conocidos son:

- **Patrón cliente-servidor:** es una arquitectura que divide un sistema en dos componentes principales: el cliente y el servidor. El cliente es responsable de solicitar y consumir servicios proporcionados por el servidor. El patrón se basa en la comunicación e interacción entre ellos.
 - Ventajas

- **Separación de responsabilidades:** el cliente se encarga de la interacción con el usuario y la presentación de los datos, mientras el servidor se encarga de la lógica, el procesamiento y el almacenamiento de datos, esto permite una mejor organización y modularidad del sistema.
- **Escalabilidad:** el modelo permite escalar y distribuir la carga de trabajo en múltiples servidores, lo cual ayuda a manejar múltiples solicitudes de forma simultánea.
- Desventajas
 - **Extrema dependencia del servidor:** si el servidor falla, los servicios que dependen de él pueden quedar inaccesibles para el cliente
 - **Necesidad de una red:** la comunicación entre el cliente y el servidor se realiza a través de una red, lo que puede introducir problemas de latencia y seguridad.
 - **Costos de implementación:** la configuración y la mantención de una infraestructura de red y servidores, puede requerir una inversión significativa.
- Casos de uso
 - Desarrollo de aplicaciones web
 - Aplicaciones de correos electrónicos
- **Patrón de capas:** es un patrón de arquitectura que organiza un sistema en varias capas lógicas o niveles de abstracción. Cada capa generada tiene una responsabilidad específica y se comunica con las capas adyacentes para generar el flujo del sistema, de esta manera se logra separar las labores de cada capa, aumentando la modularidad y la mantenibilidad del sistema.
 - Ventajas
 - **Separación de responsabilidades:** cada capa tiene una funcionalidad específica, lo que permite una mejor organización, facilitando el mantenimiento y la evolución del software.
 - **Modularidad y flexibilidad:** las capas se pueden desarrollar y modificar de forma independiente. Esto facilita la reutilización de componentes y permite cambios en una capa sin afectar a las demás.
 - **Abstracción:** cada capa encapsula su lógica y oculta los detalles internos a las capas superiores, lo que mejora la claridad y la comprensión del sistema.
 - Desventajas

- **Complejidad:** la implementación del patrón de capas puede requerir una mayor planificación y diseño inicial, ya que implica realizar una definición clara de las interfaces y la comunicación entre ellas.
 - **Riesgo de rigidez por dependencia entre capas:** si los cambios en una capa requieren modificaciones en varias capas adyacentes, puede resultar difícil y costoso realizar esas modificaciones de manera adecuada.
 - **Sobrecarga de comunicación:** la comunicación entre las capas puede generar una sobrecarga adicional, especialmente en sistemas complejos, lo que podría afectar el rendimiento.
- Tipos de Desarrollos
 - Aplicaciones empresariales
 - Sistemas de gestión de contenido
 - Aplicaciones web
- **Patrón Master-Slave:** es un patrón de arquitectura que se divide en dos partes, maestro y esclavo, cada entidad tiene su labor, en el caso de los esclavos, ellos se encargan de las labores asignadas por el maestro y este último se encarga de almacenar, procesar y representar los datos obtenidos desde los esclavos.
- Ventajas
 - **Tolerancia a fallos:** el modelo Master-Slave permite al nodo maestro distribuir las cargas de trabajo entre varios nodos esclavos, entonces en caso de que un esclavo falle, el maestro puede reasignar las tareas a otros esclavos disponibles, lo que mejora la tolerancia a fallos y garantiza la continuidad de las operaciones.
 - **Paralelismo y rendimiento:** al utilizar múltiples nodos esclavos, el sistema puede realizar tareas en paralelo, lo que aumenta el rendimiento y reduce los tiempos de respuesta.
 - **Modularidad:** los nodos esclavos pueden ser componentes independientes y reutilizables que realizan tareas específicas, lo que mejora la modularidad del sistema, facilitando el desarrollo y el mantenimiento del mismo.
- Desventajas
 - **Dependencia del nodo maestro:** si el nodo maestro falla, todo el sistema puede verse afectado y la coordinación de las tareas se interrumpe.
 - **Comunicación y latencia:** la comunicación entre el maestro y los esclavos introduce latencia, lo que puede llegar a generar un cuello de botella en el software, especialmente en sistemas distribuidos a gran escala.

- **Complejidad:** el diseño e implementación del maestro y la comunicación con los esclavos puede ser complejo, lo que requiere una planificación y gestión adecuada.
- Tipos de Desarrollos
 - Procesamiento de datos distribuidos
 - Servidores de bases de datos
 - Sistemas de búsqueda y recuperación de información
- **Patrón de eventos:** el patrón basado en eventos es una arquitectura que se enfoca en que los componentes de un sistema interactúen a través de eventos, es decir un productor genera un evento dentro del sistema y los consumidores del evento gatillado pueden devolver una respuesta o producir nuevos eventos, creando un flujo de eventos dentro del sistema.
 - Ventajas
 - **Componentes independientes:** los servicios generados en los componentes del sistema son independientes de los eventos, por lo cual genera que cada servicio creado dentro del componente tenga un comportamiento independiente de la comunicación.
 - **Flexibilidad y escalabilidad:** los eventos al ser independientes entre los componentes y los suscriptores, generan un mecanismo flexible y escalable para manejar la comunicación y la interacción entre ellos.
 - **Manejo de notificaciones y respuestas:** el patrón de eventos permite la notificación eficiente y la respuesta a eventos en tiempo real. Los componentes pueden recibir eventos relevantes y responder de manera oportuna, mejorando la capacidad de respuesta del sistema.
 - Desventajas
 - **Complejidad:** el manejo de eventos puede aumentar la complejidad del sistema, especialmente cuando se trata de eventos complejos con múltiples suscriptores y emisores.
 - **Dificultad en la depuración:** la naturaleza asincrónica del patrón de eventos puede dificultar la depuración y el rastreo de problemas en el flujo de eventos y en la respuesta de los componentes.
 - **Posibilidad de sobrecarga:** el uso excesivo de eventos puede generar una sobrecarga en el sistema, especialmente si los eventos se disparan con alta frecuencia o si hay una gran cantidad de suscriptores y emisores.
 - Tipos de Desarrollos
 - Interfaces gráficas de usuario (GUI)
 - Sistemas de mensajería y notificaciones

■ Sistemas distribuidos

- **Patrón Modelo Vista Controlador (MVC)** es un modelo que usa un conjunto de capas las cuales tienen una función específica y primordial a la hora de generar un desarrollo, en este caso existen 3 componentes fundamentales:
 - **Modelo:** es el componente que se encarga de los datos y la lógica de la aplicación.
 - **Vista:** se encarga de la interfaz de usuario, la forma en que el cliente hace uso del sistema y responde directamente a las interacciones con el usuario..
 - **Controlador:** es el componente que se encarga de recibir las solicitudes del usuario por medio de la vista, las procesa y genera la respuesta según los datos que obtiene del modelo, además mantiene el control del flujo de la aplicación y actúa como intermediario de los componentes antes mencionados.

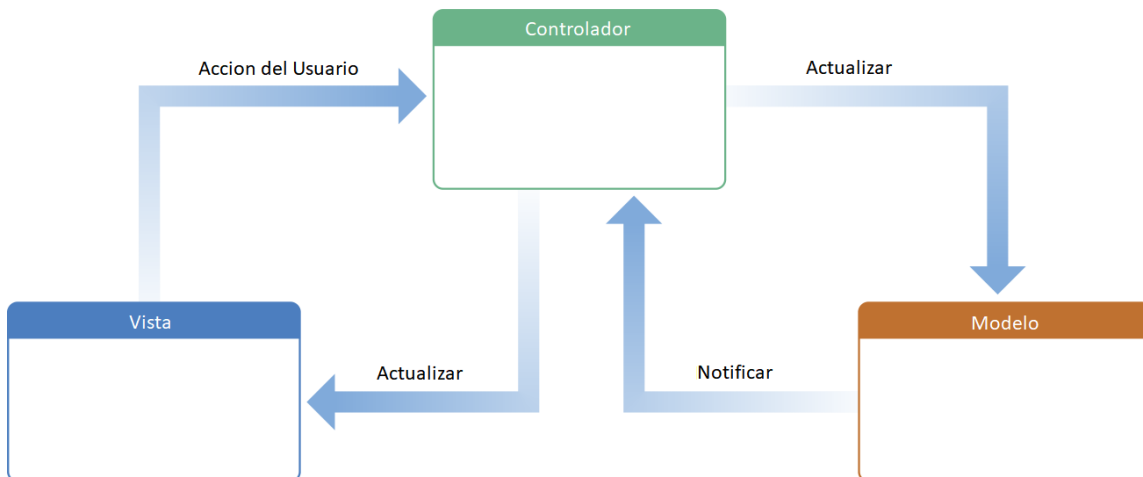


Figura 8. Esquema de patrón MVC

Fuente:

<https://developer.apple.com/library/archive/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>

En la figura 8 se puede observar cómo se realizan las interacciones entre los diferentes componentes que participan en esta arquitectura, inicialmente se ve cómo el usuario genera una acción en la vista, luego como el controlador genera un cambio dentro del modelo, el modelo notifica al controlador que el cambio fue realizado y este último realiza el cambio en la vista para que el usuario vea que su acción fue realizada, generando toda la interacción con el usuario, el sistema y la base de datos.

Al igual que los patrones anteriores, existen ventajas, desventajas y tipos de desarrollos en los cuales se aplica esta metodología.

- Ventajas
 - **Separación de responsabilidades:** al separar las partes del sistema en tres genera una clara división de las responsabilidades de cada componente, lo que facilita la organización y el mantenimiento del código.
 - **Reutilización de componentes:** al dividir la lógica de la aplicación en capas (Modelo, Vista y Controlador), se pueden reutilizar los componentes de manera independiente. Esto facilita la modularidad y la escalabilidad del sistema.
 - **Mayor mantenibilidad:** la separación de las responsabilidades en el MVC facilita las modificaciones y actualizaciones en la aplicación sin afectar a los demás componentes. Esto mejora la mantenibilidad y evita efectos colaterales indeseados.
 - **Facilidad para el testing:** al tener una clara separación de las responsabilidades, cada componente del MVC puede ser probado de forma independiente, lo que facilita la implementación de pruebas unitarias y de integración.

- Desventajas
 - **Complejidad inicial:** la implementación inicial del patrón MVC puede ser más compleja que en otros enfoques. Requiere un diseño y una estructura adecuada para asegurar una correcta interacción entre los componentes.
 - **Curva de aprendizaje:** para los desarrolladores que no están familiarizados con el patrón MVC, puede llevar tiempo y esfuerzo comprender y aplicar correctamente el patrón.
 - **Posible duplicación de código:** en algunos casos, puede surgir la necesidad de compartir código entre el Modelo y la Vista, lo que puede generar duplicación y complicar la sincronización de los cambios.

- Tipos de Desarrollos
 - Aplicaciones web
 - Aplicaciones de escritorio
 - Aplicaciones móviles

Debido a que es un patrón que permite la reutilización, la separación de responsabilidades y facilita el testing de las funcionalidades, es un excelente modelo para generar un desarrollo web en conjunto con la metodología que se elija, ya que permite realizar las pruebas del software sin interrumpir el desarrollo de otras funcionalidades.

Los patrones de arquitectura son muy relevantes al momento de generar un sistema, ya que se encargan del diseño y la estructura de la aplicación, elegir una configuración que se adecue de buena forma a los requerimientos a realizar, permiten que el desarrollo cuente con una buena base desde un inicio, logrando un buen rendimiento y disminuyendo las posibilidades de retrasos o aumentos en los costos asociados al desarrollo (Muñoz, E. A. M. Abril 2023).

2.4 Frontend

El Frontend es la parte del desarrollo web que se encarga de toda la parte visual, es decir, es la parte donde los usuarios finales observan e interactúan con la aplicación. Por medio de ella se muestra la forma en que se ingresan datos y la manera en que se muestra información solicitada por el usuario, es por esto que esta capa de desarrollo debe ser pensada cuidadosamente para que la interacción del usuario con la aplicación sea fácil, rápida e intuitiva con el objetivo de lograr una buena experiencia con el usuario.

En el área del front-end se trabaja con lenguajes mayormente del lado del cliente, los principales son

- **HTML** es un lenguaje de marcado que utiliza etiquetas para estructurar y organizar el contenido de un sitio web.
- **CSS** se encarga del formato y diseño visual de las páginas web escritas en html, permiten generar una estructura y estilo personalizado al sitio web.
- **Javascript** es un lenguaje de programación rápido y seguro que permite conectar el aspecto visual con los datos que se encuentran en una base de datos, esto ayuda a generar un dinamismo en la página web.

Los elementos antes mencionados fueron en un inicio la revolución del desarrollo web, sin embargo a medida que avanzó el tiempo, se crearon nuevas formas de trabajar, formas que permiten que el desarrollo sea más sencillo y eficiente, aquí es donde entran 2 conceptos esenciales en el área del Frontend, los famosos frameworks y las librerías.

2.4.1 Herramientas de desarrollo web

Una herramienta de desarrollo web es un software diseñado para facilitar y agilizar el proceso de creación, diseño, codificación y mantenimiento de sistemas web. Estas herramientas proporcionan un conjunto de características y funciones que ayudan a los desarrolladores a construir desarrollos de manera más eficiente y efectiva. Algunas de las labores que pueden abordar son:

- **Edición de Código:** proporcionan editores de código con resaltado de sintaxis y autocompletado para varios lenguajes de programación web, como HTML, CSS, JavaScript, PHP, etc
- **Diseño Visual:** permiten crear interfaces visuales mediante arrastrar y soltar, lo que es útil para diseñadores que no tienen habilidades de codificación intensivas.

- **Depuración y Pruebas:** ofrecen herramientas para depurar el código y probar la funcionalidad de la aplicación en diferentes navegadores y dispositivos.
- **Gestión de Proyectos:** ayudan a organizar y gestionar archivos, carpetas y recursos relacionados con el proyecto.
- **Generación Automatizada de Código:** pueden generar automáticamente partes del código repetitivo o complejo, lo que acelera el proceso de desarrollo.
- **Integración con Bases de Datos:** facilitan la conexión y manipulación de bases de datos desde la aplicación web.
- **Optimización y Rendimiento:** proporcionan herramientas para optimizar el rendimiento del sitio web, como compresión de imágenes y minificación de código.
- **Colaboración:** algunas herramientas permiten la colaboración entre equipos de desarrollo y diseñadores.

Existen varias herramientas de desarrollo web, entre ellas están los frameworks, editores de texto avanzados, herramientas de diseño, entre otros. Para el desarrollo de la solución a realizar, se va a utilizar un tipo de herramienta de desarrollo orientado al desarrollo rápido de aplicaciones web, entre ellos están.

- **OutSystems:** es una plataforma de desarrollo de aplicaciones empresariales de bajo código que permite crear aplicaciones web y móviles altamente escalables y personalizables. Se enfoca en la automatización de procesos, la integración de sistemas y la colaboración entre equipos de desarrollo y negocio.

Ventajas

- Desarrollo rápido: permite crear aplicaciones de manera rápida utilizando componentes predefinidos y código generativo.
- Automatización de procesos: facilita la creación de flujos de trabajo y la automatización de procesos empresariales complejos.
- Integración sencilla: ofrece herramientas para integrar con sistemas existentes y servicios web.
- Colaboración: fomenta la colaboración entre equipos de desarrollo y partes interesadas del negocio.
- Escalabilidad: las aplicaciones desarrolladas son altamente escalables y pueden gestionar cargas de trabajo exigentes.

Desventajas

- Costo: puede ser costoso en comparación con otras soluciones de desarrollo.
- Menos control: la naturaleza de bajo código puede limitar el control total sobre el código generado.
- Curva de aprendizaje: aunque es más rápido que el desarrollo tradicional, aún puede requerir tiempo para aprender la plataforma.

- **Scriptcase:** es una herramienta de desarrollo rápido de aplicaciones que se centra en la creación de aplicaciones web y sistemas de gestión de bases de datos. Permite generar interfaces de usuario y consultas, y es conocido por su generación automática de código.

Ventajas

- Generación automática: genera automáticamente código para formularios, consultas y otros componentes.
- Amplia compatibilidad: puede ser utilizado con varias bases de datos y tecnologías web.
- Personalización: permite personalizar los componentes generados según las necesidades.
- Rapidez: permite desarrollar aplicaciones web rápidamente, especialmente si se trabaja con bases de datos.

Desventajas

- Limitaciones de personalización: A veces puede ser limitado en la personalización avanzada de interfaces de usuario.
- Curva de aprendizaje: Puede requerir tiempo para aprender sus características y configuraciones.
- No es ideal para proyectos complejos: Puede ser menos adecuado para proyectos muy complejos o con requerimientos específicos fuera de su alcance.

- **OpenXava:** es una herramienta de desarrollo rápido de aplicaciones de código abierto para aplicaciones, la cual permite simplificar el proceso de desarrollo web y disminuir los tiempos de desarrollo.

Ventajas

- Desarrollo rápido: Permite la creación rápida de aplicaciones web utilizando anotaciones y configuraciones.
- Reutilización de código: Promueve la reutilización de componentes y módulos.
- Integración con tecnologías Java: Se integra con tecnologías y frameworks Java existentes.
- Código limpio: Genera código limpio y estructurado siguiendo patrones de diseño.

Desventajas

- Menos popularidad: Puede tener una comunidad y recursos más pequeños en comparación con otras opciones.
- Limitaciones de personalización: Puede tener limitaciones en la personalización avanzada y el diseño de interfaces.

- Curva de aprendizaje: Puede requerir cierto tiempo para aprender a usar sus características y abordar sus patrones de desarrollo.

2.4.2 Librerías JavaScript

Según Canelo, M. M. (2020, julio 8), las librerías JavaScript son un código reutilizable que a menudo tiene un caso de uso principal. Las librerías proporcionan muchas funcionalidades estándar para que los desarrolladores no tengan que preocuparse por muchas funciones genéricas, esto permite a los programadores crear páginas web fácilmente a través de los componentes de la interfaz de usuario, utilidades de lenguaje, funciones matemáticas y más.

- **JQuery:** es una librería de JavaScript rápida e incluida en un solo archivo, fue creada en 2006 y se ha convertido en una de las librerías más populares y utilizadas en la actualidad.
- **React:** es una librería de código abierto mantenida por Meta, el objetivo de ella es crear interfaces usuarias interactivas de forma rápida y sencilla.

2.4.3 User Experience (UX)

La experiencia de usuario es otro aspecto relevante a considerar al momento de generar una aplicación web, ya que se enfoca en que la interacción del usuario con el sistema sea satisfactoria, cómoda e intuitiva. Los principales aspectos en los que se enfoca la UX son:

- **Usabilidad:** se centra principalmente en que la aplicación sea fácil de usar, con el objetivo de que las tareas a realizar se hagan de forma rápida y efectiva.
- **Atractivo Visual:** se enfoca en que el diseño visual y los colores de la aplicación sean atractivos y agradables para los usuarios.
- **Accesibilidad:** su objetivo es generar una aplicación que maximice la cantidad de personas que puedan utilizarla, enfocándose en los módulos de la aplicación y la forma de interactuar con ella.
- **Arquitectura de la Información:** se centra en la organización de la información que se presenta a lo largo de la aplicación, para lograr que los usuarios puedan encontrar lo que necesitan sin dificultades.
- **Interacción del usuario:** su objetivo es generar una aplicación que le permita al usuario interactuar con el sistema de forma fluida y natural.

Lo anterior es fundamental para que una aplicación se logre consolidar dentro del

mercado y para que los usuarios se sientan atraídos por la solución y accedan frecuentemente a la aplicación.

Un punto muy importante a la hora diseñar una aplicación web son los colores, debido a que según González, M. E. (2020, julio 14) “el color nos evoca emociones y sentimientos que son difíciles de conseguir con otros recursos. Una buena paleta de colores identificará tu web como un sitio de confianza, profesional, innovador, tecnológico, dependiendo del objetivo de la marca en cuestión”, por lo cual crear una página web con colores que identifiquen a tu aplicación y a tu marca, es fundamental para que el usuario final se conecte con ella emulando ciertas sensaciones que se desean entregar a través de ella, diversas marcas utilizan estas estrategias para llegar a sus clientes algunos ejemplos son:

- **HERTZ:** es una empresa de arriendo de vehiculos que en su página web predomina el color amarillo el cual, segun la psicologia del color, provocan optimismo, energía y alegría

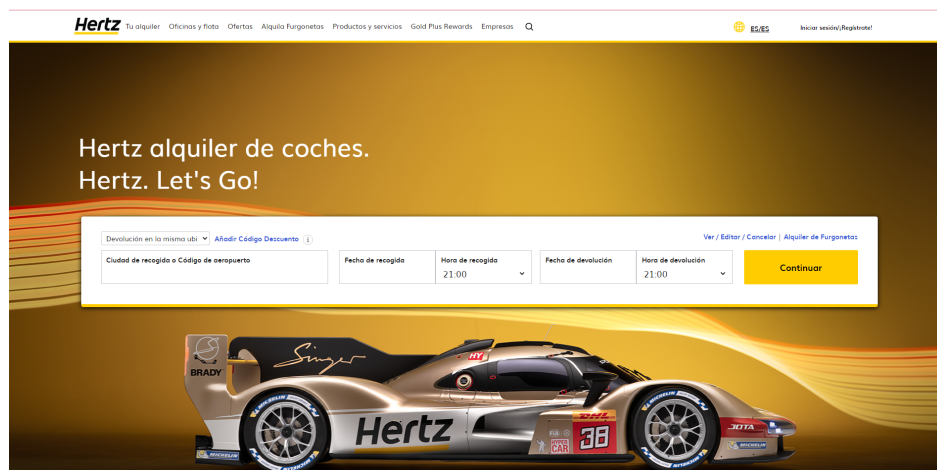


Figura 9. Elaborado por The Hertz Corporation, Página web de la empresa Hertz

Fuente: <https://www.hertz.es/rentacar/reservacion/>

- **Milka:** es una marca de dulces y chocolates, el cual en su página web y su logo predomina el color violeta, el cual según la psicología del color entrega la sensación de elegancia, fantasía e imaginación.



Figura 10. Elaborado por Milka, Página web de la empresa Milka
Fuente: <https://milka.cl/>

- **Zara:** es una marca de ropa exclusiva el cual dentro de su página web predominan 2 colores el blanco y el negro, los cuales segun la psicología del color se relacionan a la pureza y elegancia



Figura 11. Elaborado por Zara, Página web de la empresa Zara
Fuente: <https://www.zara.com/cl/>

Tal como los colores antes mencionados, los demás también representan ciertas sensaciones que le producen al usuario final, por lo cual es un aspecto relevante a tomar en consideración al momento de realizar una aplicación web.

2.5 Backend

El Back-end es el lado del desarrollo web que tiene por objetivo generar toda la lógica que existe detrás de una página web para que esta funcione sin problemas, se encarga de

ingresar, leer, editar y eliminar los datos que se encuentran guardados en la base de datos y que la comunicación con el servidor sea estable.

Las estructuras de las bases de datos son fundamentales para que el funcionamiento del servidor sea óptimo, y por consiguiente la página web sea fluida y rápida, debido a que un exceso de tiempo de respuesta del servidor genera una experiencia de usuario no deseable y un pensamiento de mal funcionamiento de la aplicación web.

Dentro del área del Back-end se encuentran muchos lenguajes de programación orientados al funcionamiento del servidor tales como:

- **PHP:** es un lenguaje de programación de código abierto ampliamente utilizado en el desarrollo web, se utiliza para la creación de aplicaciones web dinámicas y la generación de contenido en el lado del servidor, se integra con HTML y el contenido a mostrar se ejecuta primero en el servidor y luego al cliente.
 - Ventajas
 - Alto nivel de compatibilidad con los servidores.
 - Facilidad de aprendizaje.
 - Amplio nivel de soporte y documentación
 - Desventajas.
 - Necesidad de servidor web para su funcionamiento
 - No tiene herramientas de depuración ni de manejo de errores.

- **Node.JS:** es un entorno de ejecución de código abierto que permite ejecutar JavaScript en el lado del servidor por medio del motor V8 de Google Chrome, la comunicación está basada en un modelo de eventos a través de un enfoque de programación asíncrona.
 - Ventajas
 - Alto rendimiento y velocidad de ejecución.
 - Facilidad de aprendizaje para conocedores de JavaScript.
 - Amplio nivel de soporte y documentación
 - Desventajas.
 - Uso intensivo de CPU
 - No tiene herramientas de depuración ni de manejo de errores.

- **Python:** es un lenguaje de programación de alto nivel, interpretado y de propósito general. Tiene una sintaxis clara y legible, lo que facilita la escritura y la comprensión del código, además es un lenguaje multiparadigma, ya que permite la programación imperativa, orientada a objetos y funcional. Es utilizado en diferentes ámbitos desde el desarrollo web hasta la automatización de tareas y aprendizaje automático en redes neuronales.
 - Ventajas

- Sintaxis intuitiva y de fácil lectura.
- Gran cantidad de bibliotecas y módulos que facilitan el desarrollo de soluciones.
- Comunidad activa y amplia documentación.
- Desventajas.
 - Rendimiento relativamente bajo en comparación con lenguajes de más bajo nivel como C o C++
 - Alto consumo de memoria
 - Para el desarrollo de aplicaciones móviles puede estar limitado por algunas opciones y herramientas.

Por otro lado existen motores de bases de datos que se encargan de guardar y manejar los datos que son ingresados a través de la solución web, algunos de ellos son:

- **MySQL:** es un motor de bases de datos relacional de código abierto que usa el lenguaje SQL para generar consultas a la BD, es utilizado ampliamente y está mantenido y respaldado por Oracle.
 - Ventajas
 - Bajos requerimientos de hardware para su funcionamiento
 - No tiene costo asociado
 - Fácil de instalar y utilizar
 - Es escalable y de buen rendimiento.
 - Amplia comunidad y soporte técnico.
 - Desventajas.
 - En comparación con otros motores de bases de datos de pago tienen un rendimiento menor.
 - Pueden existir problemas al generar cambios en la estructura de la Base de Datos.
- **SQL Server:** es un sistema de gestión de bases de datos relacional desarrollado por Microsoft, al igual que MySQL, utiliza el lenguaje de consultas estructurado SQL para administrar y manejar la base de datos.
 - Ventajas
 - Mantiene un registro de transacciones nativo.
 - Es escalable y de buen rendimiento
 - Desventajas
 - Licenciamiento y costos asociados a la utilización del sistema.
 - Requerimientos de hardware mínimos.
 - Limitación de la edición gratuita, asociado a ciertas características y capacidad de almacenamiento

- **Postgresql:** es al igual que los anteriores es un sistema gestor de base de datos relacional de código abierto y altamente escalable. Se basa en el lenguaje SQL para manipular los datos de la base de datos y se destaca por su confiabilidad y su enfoque en la integridad de datos.
 - Ventajas
 - Alta confiabilidad.
 - Comunidad activa y con amplio soporte.
 - Desventajas.
 - Curva de aprendizaje exponencial debido a la amplia gama de características asociadas al motor.
 - Requerimientos de hardware mínimos.
 - Su ecosistema es escaso en comparación con otros motores de bases de datos.

- **Cassandra:** es un sistema de gestión de base de datos distribuido y altamente escalable, diseñado para manejar grandes volúmenes de datos y ofrecer alta disponibilidad y rendimiento. Se basa en el modelo de almacenamiento clave-valor y trabaja con una arquitectura descentralizada.
 - Ventajas
 - Ofrece una alta disponibilidad y tolerancia a fallos, debido a su arquitectura descentralizada.
 - Alto rendimiento para escritura y lectura.
 - Escalabilidad horizontal
 - Desventajas.
 - Es complejo de configurar y administrar.
 - El modelo de datos que utiliza es desnormalizado, lo cual requiere un diseño y modelado de datos más cuidadoso.

La conexión que se genera entre los lenguajes orientados al servidor y los motores de base de datos construyen la estructura del backend y por lo tanto todo lo que es la lógica que existe detrás de una aplicación web.

CAPÍTULO 3: PROPUESTA DE SOLUCIÓN

Para la solución de lo expuesto en el primer capítulo, se propone realizar una aplicación web que contemple las problemáticas mencionadas siguiendo la metodología de Programación Extrema, debido a que se necesita una arquitectura de software que permita la reutilización y la separación de responsabilidades que facilite el testing de las funcionalidades. Para ello, se va a utilizar la metodología MVC, la cual se ajusta de mejor manera a las necesidades del proyecto.

Considerando las ventajas y desventajas expuestas en el capítulo 2 respecto a los lenguajes de backend y los motores de bases de datos, la aplicación web va a utilizar el sistema:

- **Mysql:** motor de base de datos relacional.
- **PHP:** lenguaje de programación, el cual se encarga de manejar el área del servidor y realizar las solicitudes de datos a la Base de datos.

PHP y Mysql son conocidos en el área del desarrollo web y tienen un gran apoyo de la comunidad, lo cual permite que cualquier tipo de problema que se genere dentro de la aplicación se pueda enfrentar rápidamente, además de tener un excelente rendimiento sin costos asociados para el desarrollo de este, ya que son tecnologías de código abierto.

Finalmente, para el caso del frontend se va a utilizar la herramienta de desarrollo web Scriptcase, la cual permite generar formularios de forma rápida y sencilla. Además, tiene la capacidad de soportar código en HTML y JavaScript, permitiendo generar interfaces personalizadas para el cliente y utilizar el lenguaje PHP para generar solicitudes al servidor.

3.1 Implementación de la metodología Programación Extrema

Para el desarrollo de la solución Easy Key, se utilizó la metodología XP. Para su aplicación se siguieron las fases comentadas anteriormente y se realizaron reuniones con el cliente para conocer los requerimientos, ajustar los alcances del proyecto y mostrar los avances generados semanalmente.

1. Fase de planificación

Inicialmente se realizaron 2 reuniones con el cliente para conocer sus expectativas con el proyecto y sus necesidades específicas. A partir de ellas se definieron las problemáticas que afronta el cliente en sus labores diarias y se expusieron los requerimientos que el sistema debía cumplir.

Una vez definido los alcances y los requerimientos iniciales del proyecto, se comenzó a trabajar en las historias de usuario con el objetivo de formalizar y entregar al cliente las funcionalidades que el sistema iba a cumplir.

Luego, se hizo una revisión del documento entregado con el objetivo de revisar las historias de usuarios y recibir el feedback correspondiente. Se generaron las modificaciones pertinentes y el cliente aceptó la lista de historias de usuarios propuestas.

2. Fase de diseño

Una vez definidas las historias de usuario, se comenzó a seleccionar los aspectos generales del sistema, se escoge el patrón de programación, la base de datos y las herramientas de desarrollo a utilizar.

También, se definió la forma de abordar los requerimientos obtenidos en la fase de planificación, se trabajó en el diseño de la base de datos y se ordenaron las historias de usuarios para lograr una estructura de trabajo con el fin de generar entregables que permitan demostrar el avance del sistema durante cada iteración.

3. Fase de codificación

En la fase de codificación se comenzó a trabajar en las primeras historias de usuario, que tienen el propósito de recolectar datos relevantes para el administrador. Se comenzó con los formularios de registro de propietarios, luego de propiedades hasta llegar a la generación de contratos, registro de ingresos y envío de estados de cuenta. Esta estructura permitió realizar entregas al cliente final para que pueda revisar los avances presentados en cada iteración.

4. Fase de pruebas

La fase de pruebas se inició con la entrega de los primeros formularios de registro de propietarios y propiedades, esto permite que el cliente Arrecife entregue una retroalimentación a tiempo respecto a campos que se necesiten para las entidades mencionadas, como era el caso de las cuentas bancarias de los propietarios o que el sistema tuviera la capacidad de guardar datos legales de cada propiedad.

Estas pruebas permitieron que el cliente se mantuviera informado sobre los avances del sistema y que pudiera ser un participante activo del desarrollo de la solución.

5. Fase de lanzamiento

En la fase de lanzamiento, el cliente pudo revisar todas las funcionalidades del sistema y ponerlo a prueba en un entorno controlado. Se generaron una serie de pruebas exhaustivas que permiten asegurar que todas las interacciones, flujos de trabajo y características funcionaran según lo previsto. Con esto se corrigieron los últimos errores y se realizaron los ajustes finales para garantizar una experiencia de usuario fluida y sin contratiempos.

3.2 Work Breakdown Structure (WBS)

Para la organización del trabajo se realizó un Work Breakdown Structure, Estructura de Desglose del Trabajo en español. Esta es una herramienta de gestión de proyectos que descompone un proyecto grande y complejo en partes más pequeñas y manejables, lo que facilita su planificación, seguimiento y ejecución. Para el caso de un desarrollo de software la metodología de desarrollo puede influir en la estructura del WBS, debido a ello la estructura se utilizó para generar un orden inicial del proyecto para luego generar la implementación del sistema por medio de la metodología XP.



Figura 12. WBS del desarrollo de la aplicación EasyKey generado sobre una metodología XP
Fuente: elaboración propia

Como se puede observar en la figura 12, el WBS para el desarrollo de la aplicación EasyKey se separó en cuatro ramas principales, las que están directamente relacionadas con las fases de la metodología XP. La única diferencia es que la codificación y las pruebas se englobaron en la misma rama, debido a que los sprints están conformados por:

- Codificación de la interfaz de las historias de usuario.
- Codificación de la lógica de las historias de usuario.
- Testing del Sprint completo.
- Presentación del Sprint al cliente.
- Retroalimentación del Sprint.

Gracias a esto, es posible conocer lo que se debe realizar en cada sprint, las responsabilidades que tiene cada integrante del equipo y sobre todo, permite mantener el control del proyecto y da a conocer en qué faceta del desarrollo se encuentra. Como se mencionó anteriormente, el WBS está estructurado en 4 ramas principales:

- **Rama de planificación**

Como se puede observar en la figura 12, las labores de planificación se enfocan en la recopilación de información respecto a lo que debe cubrir y cumplir la aplicación a desarrollar, y cumple en general con el mismo objetivo que tiene la fase del mismo nombre en la implementación de la metodología XP.

- **Rama de diseño**

En esta rama se comienzan a revisar los aspectos globales del desarrollo, la selección de las tecnologías a utilizar, la estructura global del trabajo y el diseño de las pantallas que van a ser desarrolladas para la aplicación.

- **Rama de codificación y testing**

En la rama de codificación y testing aparecen las labores que se llevarán a cabo a nivel de código, la creación de los modelos de base de datos y el desarrollo de cada uno de los sprints, que engloban el desarrollo, el testing, la presentación y las modificaciones derivadas de la retroalimentación de las historias de usuario.

- **Rama de entrega**

Al finalizar los sprints, se llega a la última rama del WBS, que busca revisar lo desarrollado y comprobar si cumple con los alcances definidos en un inicio, proporcionando el ambiente para que el desarrollo sea cerrado y entregado.

3.3 Actores del sistema

Para realizar la solución expuesta, se deben identificar a los actores que participan en el proceso de arrendamiento:

- **Propietarios o socios**

El propietario es quien se encarga de disponer las propiedades que se arriendan a través de la solución expuesta, su interés principal es obtener una liquidación mensual con los ingresos y los gastos que genera su propiedad mensualmente.

- **Postulante o arrendatario**

Los postulantes son aquellas personas interesadas en hacer uso de una propiedad. Estos deben comunicarse con el administrador para que gestione un contrato de arriendo. En caso de que el administrador lo desee, si se genera un contrato de arriendo, el arrendatario tiene la responsabilidad de verificar que la propiedad entregada concuerde con las características expresadas en la oferta contratada, pagar el monto del arriendo mensual acordado, hacer uso de la propiedad respetando las condiciones informadas y, si el contrato lo estipula, mantener las cuentas de la propiedad al día.

- **Maestro**

El maestro es quien se encarga de realizar las modificaciones y/o arreglo dentro de una propiedad. Esto corresponde a un gasto dentro de la propiedad y no a un contrato. En caso de estar asociada al contrato, el gasto puede ser traspasado al arrendatario, dependiendo de quien tenga la responsabilidad.

- **Administrador**

Es el actor que se encarga de verificar que todas las propiedades que están a su cargo estén en perfectas condiciones y comprueba que las cuentas de cada una de ellas estén al día. Si existe un siniestro en la propiedad, el administrador tiene la facultad de dirimir sobre el responsable de este. Si el problema ocurre producto de un desastre natural o una mala mantención de la propiedad, este deberá ser revisado por el administrador y generar la solución del inconveniente.

Para el caso de EasyKey solo un actor va a interactuar directamente con el sistema, siendo el administrador quien tendrá la labor de crear a los propietarios, las propiedades, subir fotos de propiedades, entre otras acciones.

3.4 Historias de usuario

Como ya se indicó, existen varios actores que van a participar en el sistema. Sin embargo, solo el administrador o propietario de la propiedad tendrá un conjunto de funcionalidades que van a permitir el cumplimiento de los objetivos de la aplicación EasyKey. Sin embargo en caso de que el administrador no sea el propietario, este último tendrá una interacción indirecta con el sistema, ya que recibirá mensualmente su estado de cuenta a través de la aplicación y podrá conocer el estado financiero de cada una de sus propiedades.

Para la generación de las historias de usuario se deben aclarar los siguientes conceptos:

- Entidad: es una clase o concepto relevante dentro del sistema. Representa un objeto o elemento del mundo real que se desea gestionar en la base de datos. Por ejemplo, en el contexto del "Módulo CRUD Propietarios", la entidad sería "Propietarios".
- Atributos: son las características o propiedades que describen a una entidad. Cada entidad tendrá una serie de atributos que definen sus características distintivas. En el caso de la entidad "Propietarios", algunos atributos podrían ser el nombre, la dirección, el número de teléfono, entre otros.
- Operaciones CRUD: son las acciones básicas que se pueden realizar sobre una entidad en un sistema. CRUD es un acrónimo de create, read, update y delete, los cuales se definen a continuación.
 - Crear (Create): implica la capacidad de agregar nuevos registros de una entidad en la base de datos. Por ejemplo, crear un nuevo propietario y almacenar sus datos.
 - Leer (Read): permite acceder a los datos almacenados de una entidad. Puede incluir la visualización de una lista de registros de propietarios o la obtención de información detallada de un propietario en particular.
 - Actualizar (Update): consiste en modificar los datos de un registro existente de una entidad. Por ejemplo, actualizar la dirección o el número de teléfono de un propietario.
 - Eliminar (Delete): consiste en la capacidad de eliminar un registro de una entidad de la base de datos. Por ejemplo, eliminar un propietario del sistema.

Al momento de generar un requerimiento funcional puede existir la posibilidad de que ciertas entidades no puedan ser eliminadas, creadas o actualizadas. Es por esto que se va a utilizar el acrónimo según los alcances de la HU, es decir, que en caso de que una entidad no pueda ser eliminada y utilice todas las otras operaciones, entonces se va a denominar CRU de la entidad.

A continuación, se muestran las historias de usuarios del sistema a desarrollar englobados en cada sprint a desarrollar:

- **Sprint 1: Módulo de propietarios**

ID	HU-1	Título	Crear propietario
Descripción		Yo, como administrador, necesito agregar a los propietarios que deseen disponibilizar sus propiedades para poder tener sus datos en un sistema centralizado.	
Criterios de aceptación		<ul style="list-style-type: none"> ● Dado que quiero crear un propietario en el sistema, tendré que ingresar los siguientes datos obligatorios en el sistema. <ul style="list-style-type: none"> ○ Rut ○ Nombre ○ Dirección ○ Comuna ○ Email ○ Teléfono <p>Una vez rellenados, el sistema revisará los campos y los guardará en el sistema, en caso de que uno de ellos no se haya rellenado, el sistema mostrará una alerta en pantalla indicando los campos que faltan.</p>	

ID	HU-2	Título	Modificar propietario
Descripción		Yo, como administrador, necesito modificar y agregar ciertos campos de un propietario en específico para mantener una cartera de propietarios actualizada.	
Criterios de aceptación		<ul style="list-style-type: none"> ● Dado que quiero actualizar la información de un propietario en el sistema, cuando modifique los siguientes datos del propietario existente. <ul style="list-style-type: none"> ○ Rut ○ Nombre ○ Dirección ○ Comuna ○ Email ○ Teléfono ○ Estado ○ Cuentas Bancarias <p>El sistema se encargará de verificar los campos modificados y actualizará los datos del propietario, en caso de que uno</p>	

	de los campos no esté relleno, el sistema mostrará una alerta en pantalla indicando los campos que faltan.
--	--

- **Sprint 2: Módulo de propiedades**

ID	HU-3	Título	Crear propiedad
Descripción		Yo, como administrador, necesito crear las propiedades que los propietarios deseen disponibilizar en el sistema, para que puedan ser administradas y ofrecidas a posibles arrendatarios.	
Criterios de aceptación		<ul style="list-style-type: none"> ● Dado que quiero crear una nueva propiedad en el sistema, cuando complete los siguientes campos requeridos <ul style="list-style-type: none"> ○ Tipo de Propiedad ○ Rol de Propiedad ○ Comuna ○ Título de Propiedad ○ Dirección ○ Intersección Cercana ○ Estado de la Propiedad ○ Estado de Conservación <p>Una vez rellenos, el sistema revisará que los campos estén rellenos y los guardará en el sistema, en caso de que uno de ellos no se haya relleno, el sistema mostrará una alerta en pantalla indicando los campos que faltan.</p> ● Dado que necesito saber quién o quiénes son los dueños de la propiedad, cuando la registre, el sistema me permitirá asociar a los propietarios considerando el porcentaje que le corresponde de la propiedad. ● Dado que necesito complementar la información inicialmente proporcionada, cuando registre la propiedad, el sistema me permitirá adjuntar fotos en formato JPG y documentos en formato PDF, además de complementar los datos de la propiedad relleno los siguientes campos no obligatorios. <ul style="list-style-type: none"> ○ Datos Físicos <ul style="list-style-type: none"> ■ Tipo de Propiedad ■ Baños ■ Estacionamientos ■ M2 Construidos ■ Cantidad de Dormitorios ○ Gastos Recurrentes ○ Datos Legales 	

	<ul style="list-style-type: none"> ■ Nombre del Vendedor ■ Fecha de Compra ■ Fojas ○ Comercial <ul style="list-style-type: none"> ■ Valor Compra UF ■ Valor Ultimo Arriendo ■ Valor Comercial Estimado <p>Una vez rellenados, el sistema los guardará en el sistema, en caso de que algún campo contenga un error, el sistema mostrará una alerta en pantalla indicando el problema encontrado.</p>
--	---

ID	HU-4	Título	Modificar propiedad
Descripción		Yo, como administrador, necesito modificar los campos de la propiedad para mantener las propiedades que administro actualizadas.	
Criterios de aceptación		<ul style="list-style-type: none"> ● Dado que necesito actualizar la información de una propiedad, cuando modifique alguno de los campos mencionados en la HU-3, entonces el sistema actualizará los datos de la propiedad, en caso de que algún campo contenga un error, el sistema mostrará una alerta en pantalla indicando el problema encontrado. 	

● **Sprint 3: Módulo de Gastos Recurrentes**

ID	HU-5	Título	CRU Gastos programados para la propiedad
Descripción		Como administrador, necesito poder realizar operaciones CRU para los gastos programados asociados a una propiedad específica. Estos gastos programados se definen como aquellos que se repiten a lo largo del tiempo, con una temporalidad de cobro y una fecha límite de pago. Estos gastos deben crearse dentro de la entidad de gastos generados en el mes correspondiente según la temporalidad indicada por el administrador.	
Criterios de aceptación		<ul style="list-style-type: none"> ● Dado que se desea crear un nuevo gasto programado asociado a una propiedad, cuando el administrador ingrese la descripción, monto, temporalidad de cobro y fecha de replicación, entonces el sistema registrará correctamente el gasto programado, en caso de que algún campo no se haya 	

	<p>rellenado, el sistema mostrará una alerta en pantalla indicando el campo faltante.</p> <ul style="list-style-type: none"> ● Dado que se requiere consultar la información de los gastos programados existentes asociados a una propiedad, cuando el administrador acceda a la lista de gastos programados, el sistema mostrará los detalles de cada gasto, incluyendo su descripción, monto, temporalidad de cobro y fecha límite de pago. ● Dado que se desea actualizar la información de un gasto programado asociado a una propiedad, cuando el administrador modifique la descripción, monto o fechas de pago, y guarde los cambios, entonces el sistema actualizará correctamente el gasto programado. En caso de que se desee desactivar el gasto programado, este no debe generar más registros en la lista de gastos generados.
--	---

ID	HU-6	Título	Aviso de no pago de cuentas
Descripción		Como administrador, necesito recibir un aviso cuando los servicios básicos de una propiedad estén a un día de vencer, para estar informado sobre las cuentas de las propiedades a mi cargo.	
Criterios de aceptación		<ul style="list-style-type: none"> ● Dado que se desean monitorear los vencimientos de los servicios básicos de las propiedades, cuando el sistema verifique el estado de dichos servicios, como agua, luz y gas, el sistema revisará si la propiedad tiene deudas vencidas. ● Dado que se desea notificar al administrador si un servicio básico ha vencido, cuando el sistema detecte dicho vencimiento, le enviará un aviso al administrador que indique la propiedad afectada y el servicio que ha vencido. ● Dado que se busca proporcionar información útil al administrador, cuando se envíe el aviso de vencimiento de un servicio básico, el sistema incluirá detalles adicionales como la fecha de vencimiento, el proveedor del servicio y cualquier otra información relevante para facilitar la gestión de las cuentas de las propiedades. 	

● **Sprint 4: Módulo de gastos únicos**

ID	HU-7	Título	CRUD gastos Generados en el mes actual para la propiedad
----	------	--------	--

Descripción	Como administrador, necesito poder realizar operaciones CRUD para los gastos generados en el mes actual de una propiedad específica, con el fin de mantener los gastos actuales de la propiedad actualizados y fieles a la realidad.
Criterios de aceptación	<ul style="list-style-type: none"> ● Dado un nuevo gasto generado en el mes actual, cuando lo quiera registrar en el sistema, entonces se desplegará un formulario solicitando la descripción del gasto, tipo de gasto, el monto y fecha de generación. ● Dado que quiero ver los gastos generados en el mes actual de una propiedad específica, cuando acceda a la lista de gastos generados, el sistema me mostrará los gastos correspondientes al mes en curso, incluyendo su descripción, monto y fecha de generación. ● Dado que deseo modificar los gastos generados en el mes actual para una propiedad, cuando modifique alguno de los siguientes campos del gasto generado <ul style="list-style-type: none"> ○ Descripción del gasto ○ Tipo de Gasto ○ Monto ○ Fecha de generación ○ Estado <p>El sistema va a generar la modificación respectiva, asegurando que los cambios se apliquen correctamente y se reflejen en la lista de gastos generados, en caso de que algún campo contenga un error, el sistema mostrará una alerta en pantalla indicando el problema encontrado.</p> ● Dado que deseo eliminar un gasto generado en el mes actual para una propiedad que fue ingresado por equivocación, cuando seleccione el gasto que quiero eliminar, entonces el sistema lo eliminará de forma permanente, evitando su aparición en la lista de gastos generados.

ID	HU-8	Título	Módulo CRUD de pagos de gastos de las propiedades
Descripción		Como administrador, necesito poder realizar operaciones CRUD para los pagos asociados a los gastos de una propiedad específica, con el fin de mantener un registro y respaldo de los pagos realizados por gastos de la propiedad.	
Criterios de aceptación		<ul style="list-style-type: none"> ● Dado que se desea registrar un pago realizado para los gastos asociados a una propiedad, cuando el administrador 	

	<p>seleccione el gasto a pagar y suba el comprobante que corresponda, entonces el sistema registrará correctamente la información del pago.</p> <ul style="list-style-type: none"> • Dado que se desea consultar el estado de pago del listado de gastos asociados a una propiedad, cuando el administrador acceda a la lista de gastos de la propiedad, el sistema mostrará detalles de los gastos con sus estados correspondientes, incluyendo el monto, la fecha del gasto, la fecha de pago y los detalles de la transacción. Adicionalmente, en la parte superior izquierda de la lista de gastos, se presentará un filtro que permitirá al administrador ordenar y filtrar los gastos según la fecha o descripción. • Dado que necesito actualizar o corregir la información de un pago registrado cuando realice la modificación del pago correspondiente, entonces el sistema actualizará la información del pago, asegurando la integridad de los registros del sistema, en caso de que algún campo contenga un error, el sistema mostrará una alerta en pantalla indicando el problema encontrado. • Dado que necesito eliminar un pago mal ingresado, cuando el administrador suprima el pago seleccionado, entonces el sistema eliminará el registro de forma permanente, asegurando la integridad de los registros de pagos.
--	--

• **Sprint 5: Módulo de Postulantes**

ID	HU-9	Título	Crear postulante
Descripción		Como administrador, necesito crear un postulante para alimentar una cartera de posibles arrendatarios y que estos puedan ingresar al sistema.	
Criterios de aceptación		<ul style="list-style-type: none"> • Dado que se desea crear un nuevo postulante en el sistema, cuando el administrador ingrese los siguientes datos personales del postulante <ul style="list-style-type: none"> ○ Rut ○ Nombre ○ Comuna ○ Dirección ○ Correo ○ Teléfono ○ Tipo de Empleo ○ Cargo 	

	<ul style="list-style-type: none"> ○ Ingresos Mensuales <p>entonces el sistema lo registrará en la base de datos, en caso de que falte un campo por rellenar, el sistema mostrará una alerta en pantalla indicando los campos faltantes.</p>
--	---

ID	HU-10	Título	Modificar datos del postulante
Descripción		Como administrador, necesito modificar los datos de un postulante para mantener los datos del postulante actualizados y fieles a la realidad.	
Criterios de aceptación		<ul style="list-style-type: none"> ● Dado un postulante, cuando el administrador desee modificar los siguiente datos del postulante <ul style="list-style-type: none"> ○ Rut ○ Nombre ○ Comuna ○ Dirección ○ Correo ○ Teléfono ○ Tipo de Empleo ○ Cargo ○ Ingresos Mensuales <p>entonces el sistema desplegará un formulario para que reemplace los datos que desee modificar y pueda guardarlos.</p> <ul style="list-style-type: none"> ● Dado que necesito complementar los datos del postulante con información de fuentes externas, cuando necesite adjuntar antecedentes, el sistema me permitirá adjuntar fotos en formato JPG y documentos en formato PDF para registrarlos dentro del sistema. 	

● **Sprint 6: Módulo de generación de contratos**

ID	HU-11	Título	Generar un precontrato tipo de arriendo a un postulante
Descripción		Como administrador, necesito crear y adjuntar un pre contrato de arriendo para un postulante para una propiedad específica, para que el cliente tenga la información del acuerdo antes de realizar la firma del acuerdo.	
Criterios de aceptación		<ul style="list-style-type: none"> ● Dado que necesito generar un precontrato para una propiedad específica en el sistema para un postulante a un bien raíz, cuando el administrador seleccione la propiedad a 	

	<p>arrendar deberá seleccionar al postulante y rellenar los siguientes campos</p> <ul style="list-style-type: none"> ○ Fecha inicio de Contrato ○ Dia del Cobro ○ Gastos Legales ○ Duración del contrato ○ Valor del arriendo ○ Valor de la garantía <p>Ademas, podre adjuntar un PDF con las condiciones del precontrato para que el postulante lo pueda firmar, entonces el sistema lo registrará en la base de datos y se encargará de enviar vía mail el documento que informa las condiciones del acuerdo al postulante.</p>
--	---

ID	HU-12	Título	Generar contrato de arriendo final con un postulante para una propiedad específica
Descripción		Como administrador, necesito poder iniciar un contrato de arriendo con un postulante previamente informado y subir el contrato firmado por las partes en formato PDF que respalde el acuerdo. Se deben tener en cuenta los plazos de validez del contrato y sus condiciones.	
Criterios de aceptación		<ul style="list-style-type: none"> ● Dado que se requiere iniciar un arriendo, cuando reciba la respuesta del postulante respecto a un precontrato previamente enviado, podre seleccionar el precontrato y modificar los siguientes campos en caso de ser necesario <ul style="list-style-type: none"> ○ Fecha inicio de Contrato ○ Dia del Cobro ○ Gastos Legales ○ Duración del contrato ○ Valor del arriendo ○ Valor de la garantía <p>Además, tendré la capacidad de adjuntar el documento firmado por el postulante en formato PDF. Una vez se realicen las modificaciones pertinentes y se adjunte el documento firmado, el contrato de arriendo entre las partes se dará por iniciado de manera oficial.</p>	

ID	HU-13	Título	Revisión detallada del contrato
----	-------	--------	---------------------------------

Descripción	Como administrador, necesito revisar una ficha con los datos del contrato seleccionado, donde pueda ver las deudas de cada propiedad, para mantener el control de la propiedad y poder realizar labores de cobranza a tiempo.
Criterios de aceptación	<ul style="list-style-type: none"> • Dado que se desea revisar la ficha de datos de un contrato seleccionado, cuando el administrador acceda a la ficha de datos del contrato, entonces el sistema mostrará la información detallada del contrato elegido. • Dado que se quiere mantener el control de la propiedad, cuando el administrador revise la ficha de datos del contrato, entonces el sistema mostrará los cobros generados por el administrador y que no han sido pagados por el arrendatario. • Dado que se necesita realizar labores de cobranza a tiempo, cuando el administrador identifique cobros pendientes en la ficha de datos del contrato, entonces el administrador podrá por medio del sistema generar notificaciones o recordatorios para que el arrendatario esté en conocimiento de su cobros no pagados.

• **Sprint 7: Módulo de cobros asociados al arriendo**

ID	HU-14	Título	Generación de cobros por arriendo
Descripción		Como administrador, necesito que el sistema genere los cobros por concepto de arriendo a los contratos que se encuentran activos, para conocer que arriendos tengo que cobrar mensualmente	
Criterios de aceptación		<ul style="list-style-type: none"> • Dado que necesito conocer los arriendos que debo cobrar en el mes en curso, cuando inicie un nuevo mes, el sistema realizará automáticamente la carga de los cobros asociados a cada contrato activo por concepto de arriendo. Esta operación incluirá la identificación del titular a cargo de cada contrato, los montos a pagar y la fecha acordada entre las partes para el cobro. 	

ID	HU-15	Título	Generar cobros extras al arrendatario
----	-------	--------	---------------------------------------

Descripción	Como administrador, necesito generar cobros extras asociados al contrato activo de una propiedad específica, para tener un registro de lo que le tengo que cobrar al arrendatario.
Criterios de aceptación	<ul style="list-style-type: none"> • Dado que se desea generar cobros asociados al contrato activo de una propiedad específica al arrendatario a cargo del contrato, cuando el administrador seleccione el contrato al cual desea generar el cobro, el administrador deberá indicar el monto, describir los aspectos relevantes del cobro y guardar el registro, entonces el sistema lo creará, lo asociará al contrato correspondiente en la plataforma.

ID	HU-16	Título	Registrar pago asociado al cobro a un arrendatario
Descripción	Como administrador, necesito registrar el pago asociado a un cobro generado a un arrendatario para mantener un respaldo de los pagos realizados por el arrendatario.		
Criterios de aceptación	<ul style="list-style-type: none"> • Dado que el administrador desea mantener todos los pagos de los arrendatarios registrados en el sistema, cuando llegue un pago, el administrador podrá buscarlo en el sistema y registrar el pago en formato JPG o PDF para tener un respaldo del mismo. 		

• **Sprint 8: Módulo de maestros**

ID	HU-17	Título	CRU de Maestros
Descripción	Como administrador, necesito realizar operaciones CRU para la entidad de maestros, con el fin de generar una lista de maestros de obras de mi confianza		
Criterios de aceptación	<ul style="list-style-type: none"> • Dado que se desea crear nuevos maestros de obras, cuando el administrador ingrese su información personal y de contacto en el sistema, entonces el sistema registrará correctamente al maestro de obras en la entidad correspondiente. • Dado que se necesita acceder a la información de los maestros de obras existentes, cuando el administrador consulta la lista de maestros de obras, el sistema muestra los datos personales del maestro y sus áreas de especialización. • Dado que se requiere mantener actualizada la información de los maestros de obras, cuando el administrador realice cambios en los datos de contacto o actualice la información 		

	personal de algún maestro de obras, el sistema guardará los cambios en la base de datos..
--	---

● **Sprint 9: Módulo de labores**

ID	HU-18	Título	CRUD labores de maestranza
Descripción		Como administrador, necesito realizar operaciones CRUD para las labores de maestranza de una propiedad en particular, con el fin de mantener un registro de los arreglos de cada propiedad y los gastos asociados a estos.	
Criterios de aceptación		<ul style="list-style-type: none"> ● Dado que se desea registrar una nueva labor de maestranza, cuando el administrador seleccione una propiedad, describa la labor a realizar y suba las imágenes en formato PDF (opcional), entonces el sistema registrará la labor en la tabla correspondiente. ● Dado que se requiere mantener actualizada la información de las labores de maestranza, cuando el administrador realice cambios en la descripción o en los datos relevantes de la labor, entonces el sistema actualizará correctamente los registros. ● Dado que se requiere eliminar una labor que ya no es necesaria, cuando el administrador seleccione la labor y la elimine, entonces el sistema eliminará la labor correspondiente. 	

ID	HU-19	Título	Asignar labores de maestranza a los maestros
Descripción		Como administrador, necesito asignar las labores de maestranza a los maestros registrados en la plataforma, para tener un listado de las labores desarrolladas por cada maestro.	
Criterios de aceptación		<ul style="list-style-type: none"> ● Dado que se desea asignar una labor de maestranza a un maestro registrado, cuando el administrador seleccione la labor de maestranza y le asigne un maestro, entonces el sistema registrará la asignación de la labor al maestro correspondiente. ● Dado que se necesita mantener un registro de las labores de maestranza asignadas, cuando el administrador quiera visualizar el listado de labores asignadas, el sistema mostrará 	

	las labores con sus estados correspondientes y el nombre del maestro al cual está asociada esa labor.
--	---

ID	HU-20	Título	Revisar labores de maestranza asignada
Descripción		Como administrador, necesito revisar las labores de maestranza asignadas a maestros registrados en la plataforma, para tener un listado de las labores desarrolladas por cada maestro.	
Criterios de aceptación		<ul style="list-style-type: none"> • Dado que requiero consultar las labores de maestranza asignadas, cuando el administrador quiera visualizar las labores asignadas, el sistema mostrará las labores con sus estados correspondientes y el nombre del maestro asociado a esa labor. 	

ID	HU-21	Título	Ingresar gastos asociados a una labor específica
Descripción		Como administrador, necesito ingresar los gastos que están asociados a las labores de maestranza que son ejecutadas por los maestros, para tener un listado actualizado de los gastos asociados a cada propiedad por concepto de labores de maestranza.	
Criterios de aceptación		<ul style="list-style-type: none"> • Dado que se desea registrar los gastos de una labor específica, cuando el administrador seleccione dicha labor, él podrá ingresar los gastos asociados a la misma. El sistema deberá guardar y registrar los gastos como parte del mes que el administrador ingresó al momento de generar el gasto. • Dado que se desea mantener los gastos mensuales de cada propiedad, cuando el administrador ingrese un gasto asociado a una labor, entonces el sistema se encargará de registrarlo junto a los otros datos generados durante el mes correspondiente. 	

Sprint 10: Modulo de estados de cuenta

ID	HU-22	Título	Revisión y envío de las liquidaciones de los propietarios
Descripción		Como administrador, necesito revisar, pagar y enviar las liquidaciones generadas a los propietarios de las propiedades administradas. Esto me permite cobrar mis ganancias	

	correspondientes o pagar la diferencia al propietario en caso de ser necesario.
Criterios de aceptación	<ul style="list-style-type: none"> • Dado que se desea revisar las liquidaciones de los propietarios, cuando el administrador acceda al listado de liquidaciones generadas por el sistema, el sistema deberá mostrar un resumen que incluya los gastos totales, los ingresos generados y el balance correspondiente, junto con la comisión del administrador. • Dado que se necesita conocer el detalle de una liquidación específica, cuando el administrador seleccione una liquidación y acceda a su detalle, el sistema deberá mostrar todos los gastos e ingresos generados en el mes correspondiente al documento de liquidación. Esto debe considerar todas las propiedades del propietario asociadas a la liquidación, así como la comisión que le corresponde al administrador. • Dado que existe la posibilidad de que los ingresos totales de una liquidación superen los gastos totales, cuando el administrador realice el pago de la liquidación y suba el comprobante en formato JPG o PDF, el sistema deberá habilitar un botón para enviar la liquidación al propietario a su correo correspondiente.

• **Sprint 11: Módulo de Informes e historiales**

ID	HU-23	Título	Historial y gestión de cobros al arrendatario
Descripción			Como administrador, necesito tener un historial de los cobros realizados a los arrendatarios por cada contrato de arriendo, ordenado por fecha desde la más actual a la más antigua, para tener un panorama general de los cobros pagados y pendientes de cada propiedad.
Criterios de aceptación			<ul style="list-style-type: none"> • Dado que se necesita tener una visión clara de los cobros pendientes, cuando el administrador revise el historial de cobros, el sistema mostrará los cobros en orden cronológico, comenzando por los cobros más recientes, lo que permite

	<p>identificar fácilmente los cobros pendientes y tener una visión general de la situación financiera de la propiedad.</p> <ul style="list-style-type: none"> ● Dado que se requiere conocer los detalles de cada cobro, cuando el administrador seleccione un cobro en el historial, el sistema mostrará la información relevante del mismo, como la propiedad, el monto, la fecha de vencimiento y el estado del cobro (pagado, pendiente, en revisión, vencido), brindando una visión completa y detallada de los cobros realizados.
--	--

ID	HU-24	Título	Historial de gastos y su detalle
Descripción	Como administrador, necesito revisar un historial de gastos de cada propiedad, ordenado por fecha del más actual al más antiguo, para tener una visualización real de los últimos gastos de la propiedad seleccionada.		
Criterios de aceptación	<ul style="list-style-type: none"> ● Dado que se desea acceder al historial de gastos de cada propiedad, cuando el administrador seleccione la propiedad específica, entonces el sistema mostrará el historial de gastos asociado a esa propiedad. ● Dado que se requiere visualizar los gastos en orden cronológico, cuando el administrador acceda al historial de gastos de la propiedad seleccionada, entonces el sistema mostrará los gastos en orden descendente según su fecha, comenzando por los gastos más recientes. ● Dado que se necesita obtener información detallada sobre cada gasto, cuando el administrador revise el historial de gastos de la propiedad, entonces el sistema proporcionará información como la descripción del gasto, el monto, la fecha y la propiedad asociada a cada registro de gasto, además de poder revisar el comprobante de pago en caso de haber registrado el pago dentro del sistema. 		

ID	HU-25	Título	Historial de contratos
Descripción	Como administrador, necesito consultar un historial de contratos, ordenado por fecha del más actual al más antiguo, para tener los datos de los arrendatarios que han hecho uso de la propiedad y tener un panorama de la propiedad a lo largo del tiempo.		

Criterios de aceptación	<ul style="list-style-type: none"> • Dado que se desea consultar el historial de contratos de una propiedad, cuando el administrador acceda al historial de contratos, entonces el sistema mostrará los contratos ordenados por fecha de manera descendente. • Dado que se quiere obtener los datos de los arrendatarios que han hecho uso de la propiedad, cuando el administrador seleccione un contrato en el historial, entonces el sistema mostrará la información relevante de ese contrato, como el arrendatario, la fecha de inicio y fin del contrato, los términos y condiciones acordados, además se mostrarán las imágenes correspondientes a cuando se entrega la propiedad al arrendatario y cuando se da por finalizado el arriendo
--------------------------------	--

ID	HU-26	Título	Informe de ganancias del administrador
Descripción		Como administrador, necesito generar un informe que contenga todos los ingresos totales mensuales del administrador de EK, considerando todas las liquidaciones generadas hasta el momento.	
Criterios de aceptación		<ul style="list-style-type: none"> • Dado que se desea generar un informe de ingresos totales del administrador, cuando el administrador ingrese al informe, entonces el sistema generará un resumen de todas las liquidaciones generadas, con los ingresos, los gastos, el balance y la comisión total de cada mes 	

ID	HU-27	Título	Informe de gestión EK
Descripción		Como administrador, necesito generar un informe que recopile todos los movimientos pendientes en el sistema durante el mes, tomando en cuenta todas las propiedades, sus cobros pendientes y los gastos que no han sido pagados a través del sistema, para poder tener una visión general y real de mi gestión.	
Criterios de aceptación		<ul style="list-style-type: none"> • Dado que se desea generar un informe de gestión, cuando el administrador ingrese al informe de gestión, entonces el sistema mostrará un informe con todos los movimientos que no han sido cerrados en el sistema hasta el momento, mostrando la fecha del movimiento. 	

3.5 Base de datos

La base de datos a utilizar para generar la solución propuesta es el motor relacional MySQL, debido a su buen rendimiento, su bajo costo y su compatibilidad con diferentes lenguajes de programación. Para explicar la base de datos propuesta, se generó una variación de un diagrama de clases el cual presenta las diez macro entidades del sistema, que engloban varias tablas de la base de datos y la tabla principal de la macro entidad.

En el siguiente esquema no se muestran los atributos de cada entidad debido al espacio disponible, permitiendo una mejor visualización de la figura.

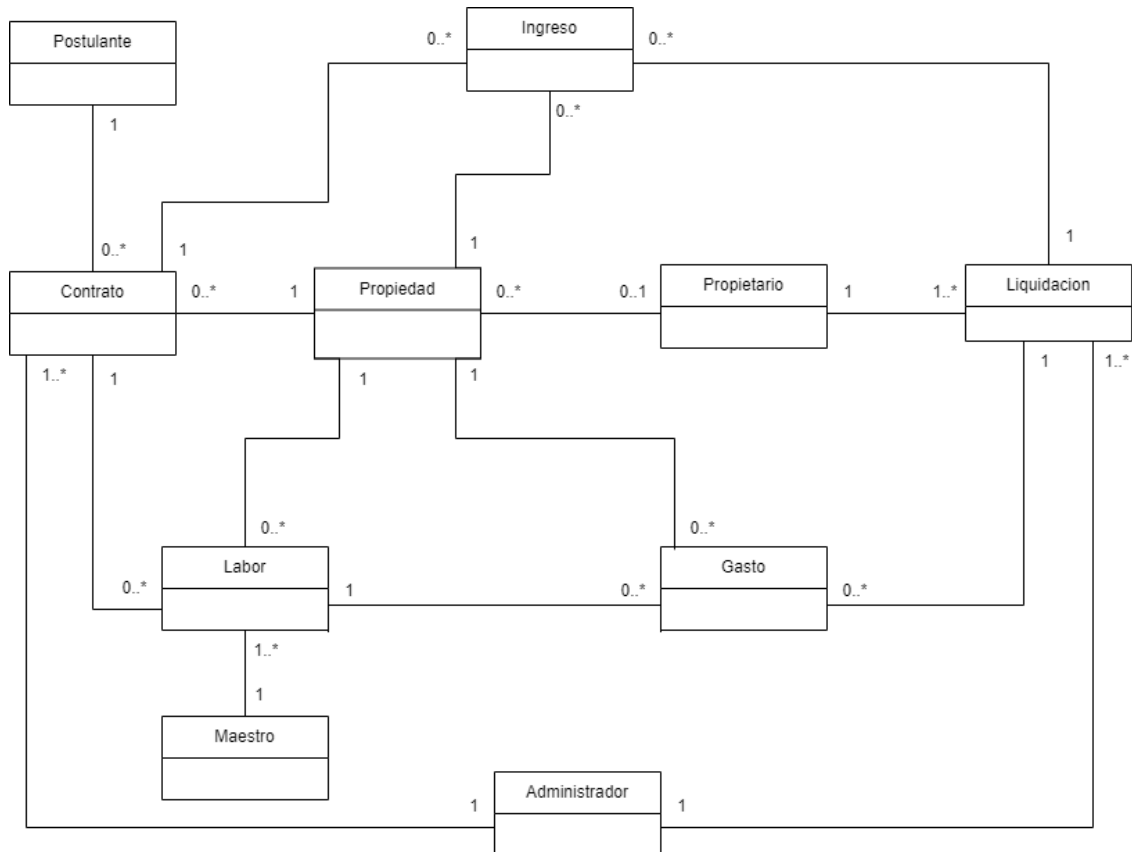


Figura 13: Diagrama ER simplificado del sistema Easy Key

Fuente: elaboración propia.

En la figura 13, se presentan las principales entidades del sistema Easy Key con sus relaciones y las cardinalidades correspondientes. Las cardinalidades se representan mediante cuatro tipos:

- **0..1**: indica que puede existir cero o una única entidad asociada a otra entidad.
- **1**: indica que hay una única entidad asociada a otra entidad.
- **1..***: indica que hay una o múltiples entidades asociadas a otra entidad.
- **0..***: indica que puede haber cero o múltiples entidades asociadas a otra entidad.

Estas cardinalidades permiten comprender las relaciones entre las entidades. Por ejemplo, la entidad "Propietario" tiene dos relaciones. La primera relación se establece con la entidad "Liquidación", y su cardinalidad indica que un propietario puede tener una o múltiples liquidaciones, mientras que cada liquidación está asociada a un único propietario. No existe una liquidación que esté asociada a varios propietarios distintos. Por otro lado, la segunda relación se genera con la "Propiedad" y, según la cardinalidad, indica que un propietario puede tener cero o múltiples propiedades asociadas a él. Sin embargo, cada propiedad debe tener solo un propietario asociado.

El objetivo del sistema es lograr una administración óptima de las propiedades. Estas propiedades son ingresadas por un administrador del sistema, debido a que los propietarios no tienen la capacidad o el interés de administrar sus bienes raíces. Por lo tanto, la entidad "Propietario" es fundamental, ya que a partir de ellos se derivan otras entidades como "Propiedad", "Liquidación", entre otras.

Los propietarios entregan el control de las propiedades para su arriendo al administrador del sistema. Este administrador se encarga de revisar los gastos e ingresos de cada propiedad a través del sistema. Esta tarea se realiza a cambio de un porcentaje de los ingresos generados por cada propiedad durante el mes. Estos detalles se reflejan en la liquidación que se genera mensualmente para cada propietario que ha entregado su propiedad para su administración.

Las liquidaciones, como se muestra en la figura 13, se generan en base a los ingresos y egresos generados por todas las propiedades de un propietario en particular. Los gastos están asociados a diferentes conceptos, como gastos comunes, cuentas y pagos de contribuciones específicas de cada propiedad. Sin embargo, existe un tipo de gasto adicional relacionado con las labores realizadas en una propiedad, como reparaciones, mantenimiento o incidentes que puedan ocurrir durante el arrendamiento. Estas labores están específicamente asociadas a una propiedad y pueden ser solicitadas por un arrendatario con un contrato activo. En ese caso, el administrador tiene la responsabilidad de gestionar la reparación o rechazar la solicitud, según corresponda. Las labores son ejecutadas por un maestro previamente registrado, el cual se encarga de realizar todo lo que tiene que ver con la tarea a realizar, desde la mano de obra, hasta los materiales que se necesitan. Los gastos generados por la tarea son informados al administrador y este último se encarga de registrar los datos en el sistema.

Por otro lado, los ingresos de las propiedades están relacionados con los pagos de arriendo generados por contratos específicos vinculados a cada propiedad. Sin embargo, también pueden generar ingresos adicionales debido a reembolsos de gastos previamente pagados u otras fuentes de ingresos no directamente relacionadas con un contrato en particular.

Los contratos de arriendo son generados entre un postulante y el administrador de bienes raíces por una propiedad en particular. Este contrato cuando se confirma y se firma entre

las partes, le permite al postulante realizar solicitudes de arreglo y generar los pagos de arriendo, lo cual genera los gastos e ingresos correspondientes.

Los postulantes son los posibles arrendatarios de una propiedad, dentro del sistema son registrados por el administrador para que puedan crearse los contratos de arriendo antes mencionados, cada uno tiene características que los diferencian y tienen la capacidad de ingresar al sistema.

A continuación, se presentan las macro entidades en detalle, revisando las relaciones internas de cada una de ellas:

3.5.1 Propietario

La macro entidad “Propietario” registra a los propietarios que entregan sus propiedades para su administración, que son ingresados al sistema por medio del administrador del software, la entidad principal tiene en total nueve atributos, que entregan al administrador datos relevantes para la comunicación con ellos.

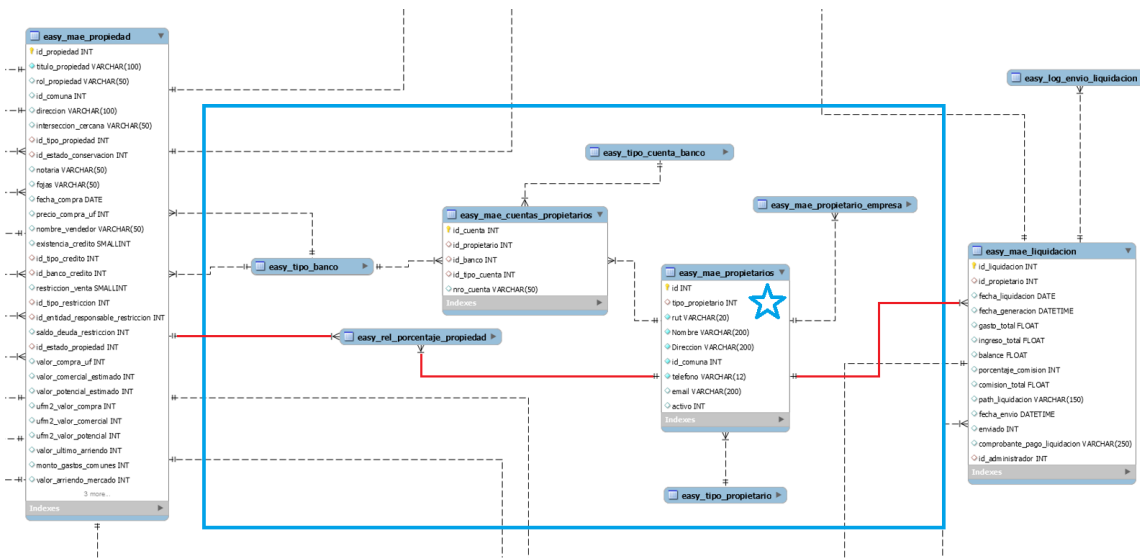


Figura 14. Diagrama ER que detalla la macro entidad “Propietarios” y se destaca en rojo las relaciones con las entidad principal de las otras macro entidades.

Fuente: elaboración propia

En la figura 14, se resaltan en color rojo las relaciones de la entidad "Propietario" con las entidades principales de la macro entidad "Propiedad" y "Liquidación". También se pueden observar otras entidades que participan con la tabla principal de propietarios, como sus cuentas y los bancos a los que pertenecen. Además, se considera el tipo de propietario, el cual puede ser una persona natural o una empresa. En el caso de que el propietario sea una empresa, se establece una relación con la tabla de propietarios de empresa. Esta tabla permite registrar a los socios de la empresa que actúa como

3.5.4 Maestro

La macro entidad "Maestro" juega un rol crucial en la gestión de los profesionales que colaboran con el administrador del sistema. Su función principal es crear y mantener una cartera de personas disponibles para ser contratadas con el propósito de llevar a cabo labores específicas en las propiedades que lo requieran. Los maestros registrados en el sistema poseen habilidades y experiencia diversas para realizar distintas tareas necesarias en las propiedades administradas por el sistema. La tabla principal de la macro entidad cuenta con ocho atributos que permiten almacenar información como el nombre del maestro, sus datos de contacto y las relaciones que mantiene con otras tablas.

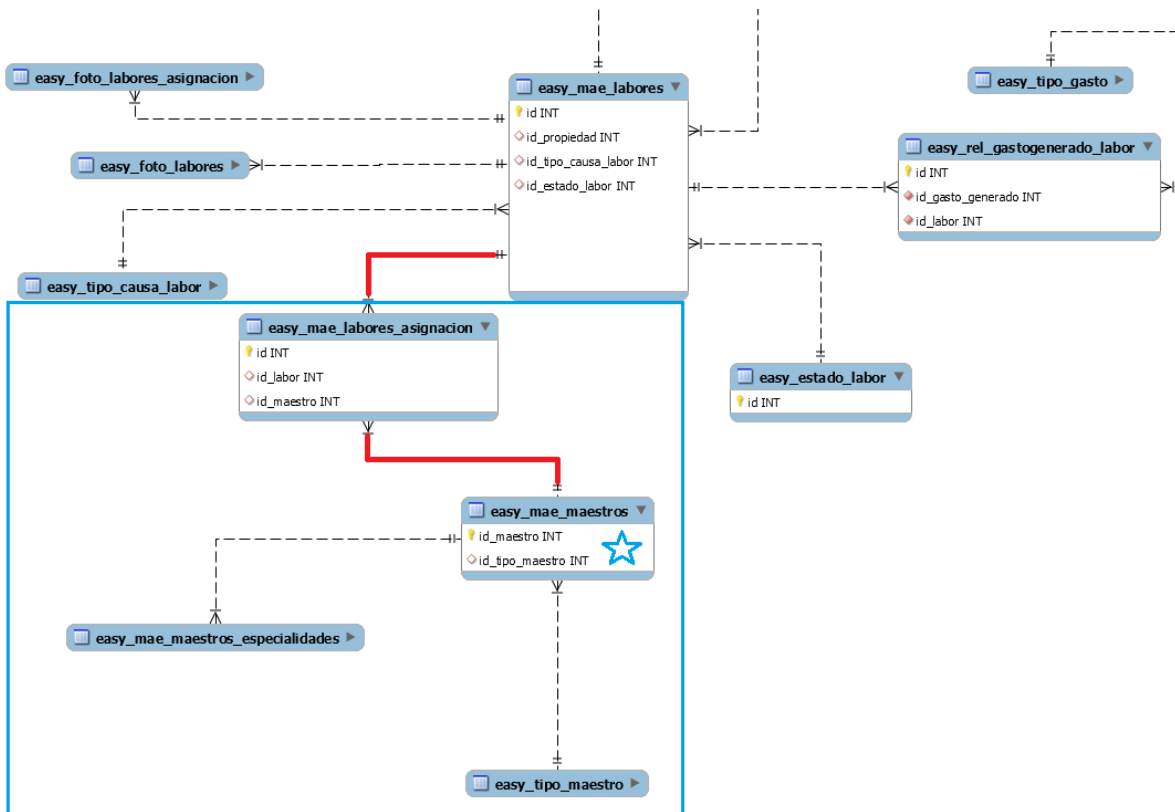


Figura 18. Detalle de la macro entidad "Maestro" donde se reflejan en rojo las relaciones principales y las otras relaciones que la complementan.

Fuente: elaboración propia

La figura 18 muestra la tabla principal de la macro entidad "Maestro", resaltada con una estrella de color azul. Las relaciones marcadas en rojo representan las conexiones que esta entidad tiene con la tabla principal de la macro entidad "Labor". Además, se incluyen las tablas auxiliares que permiten registrar las especialidades de cada maestro y su área de trabajo. Estas tablas auxiliares permiten categorizar a los maestros según sus habilidades y especialidades, como electricista, plomero, constructor, entre otras opciones. Esto facilita

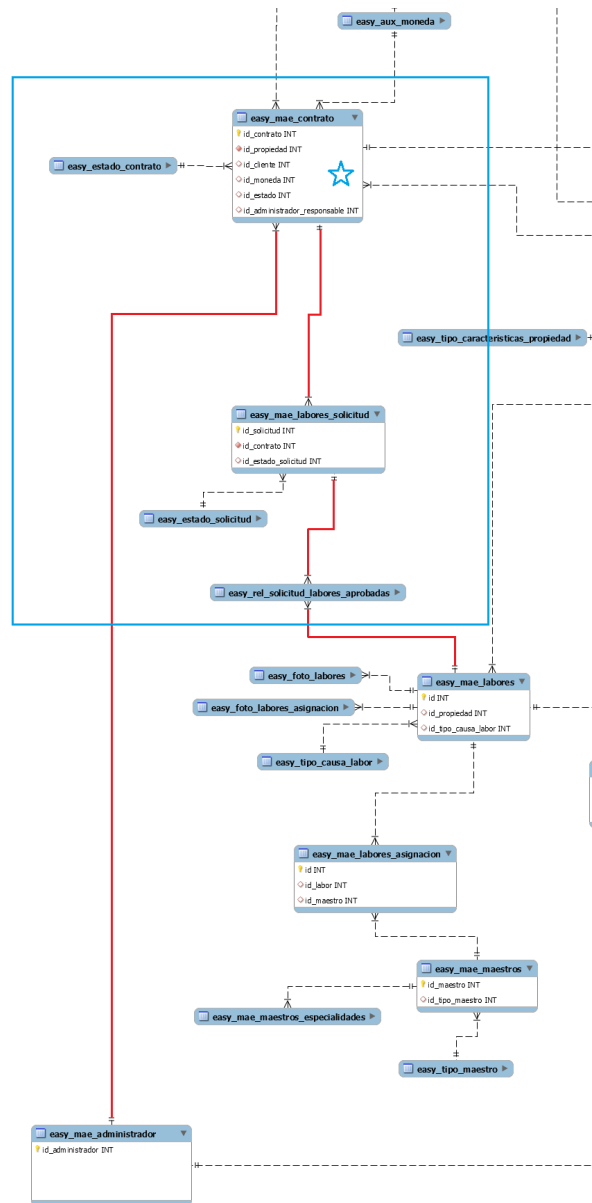


Figura 21. Diagrama ER de la segunda parte del detalle la macro entidad "Contratos" y se destaca en rojo las relaciones con la entidad principal de las otras macro entidades.

Fuente: elaboración propia

En la figura 21, al igual que en la figura 20, se presenta la tabla principal de la macro entidad "Contratos", resaltada con una estrella azul. Sin embargo, en la figura 21 se incluyen las tablas principales de las otras macro entidades relevantes "Administrador" y "Labores". Estas relaciones permiten tener el control sobre el administrador responsable del contrato y registrar las labores solicitadas por el arrendatario, así como el estado de cada solicitud. Esto brinda al sistema la capacidad de generar una trazabilidad de los trabajos realizados en la propiedad durante un contrato vigente.

Todo esto permite realizar un seguimiento detallado de todas las actividades y servicios relacionados al contrato creado.

3.5.7 Ingresos

La macro entidad "Ingresos" tiene la responsabilidad de mantener un registro de los ingresos generados por una propiedad. La tabla principal de esta entidad cuenta con 8 atributos que permiten almacenar los aspectos más relevantes de un ingreso, como la descripción, el monto, la fecha, el comprobante y el tipo de ingreso. Estos atributos son fundamentales para conocer el origen de los datos ingresados y garantizar la correcta contabilización del ingreso en la cuenta del administrador del sistema.

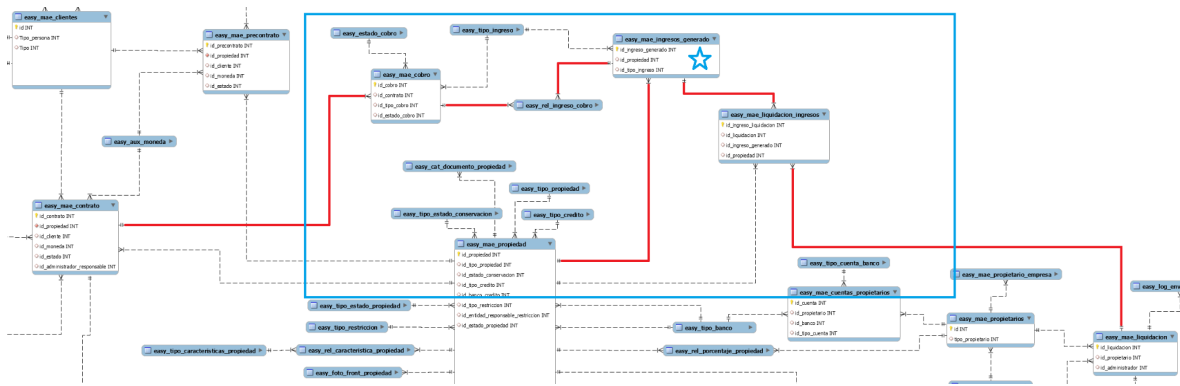


Figura 22. Detalle de la macro entidad "Ingresos" donde se reflejan en rojo las relaciones principales y las otras relaciones que la complementan.

Fuente: elaboración propia

En la figura 22, se presenta el detalle de la macro entidad "Ingresos" y su tabla principal, destacada con una estrella de color azul. Las relaciones marcadas en rojo representan las conexiones que esta entidad tiene con las tablas principales de las macro entidades "Liquidaciones", "Propiedades" y "Contratos". Estas relaciones establecen los vínculos entre los ingresos generados en una propiedad durante la vigencia de un contrato, lo cual permite registrarlos adecuadamente y generar la liquidación correspondiente al mes en el que se produjo el ingreso.

Las relaciones mencionadas en el diagrama permiten una integración eficiente de la información relacionada con los ingresos, facilitando un seguimiento completo de los ingresos generados en las propiedades administradas por el sistema y su correspondiente asociación con los contratos y las liquidaciones mensuales.

3.5.8 Postulante

La macro entidad "Postulante" desempeña un papel crucial en el registro y seguimiento de los postulantes y arrendatarios del sistema. La tabla principal de esta entidad consta de 28

atributos que permiten almacenar diversos datos necesarios para establecer contacto y evaluar la confiabilidad de una persona al momento de generar un contrato de arrendamiento. Algunos de los campos relevantes incluyen el RUT, nombre, tipo de persona (ya sea natural o jurídica), ingresos mensuales y datos de contacto, como correo electrónico y número de teléfono. También se registran datos como ocupación y otros detalles pertinentes.

Los campos mencionados son de gran importancia para evaluar a la persona y reducir la posibilidad de enfrentar problemas durante un contrato de arriendo, especialmente aquellos relacionados con la situación económica del arrendatario. Al contar con esta información detallada, el sistema puede realizar una evaluación más completa de los potenciales arrendatarios y tomar decisiones informadas al momento de celebrar un contrato de arriendo. Esto ayuda a garantizar una mayor seguridad y confiabilidad en las transacciones realizadas dentro del sistema.

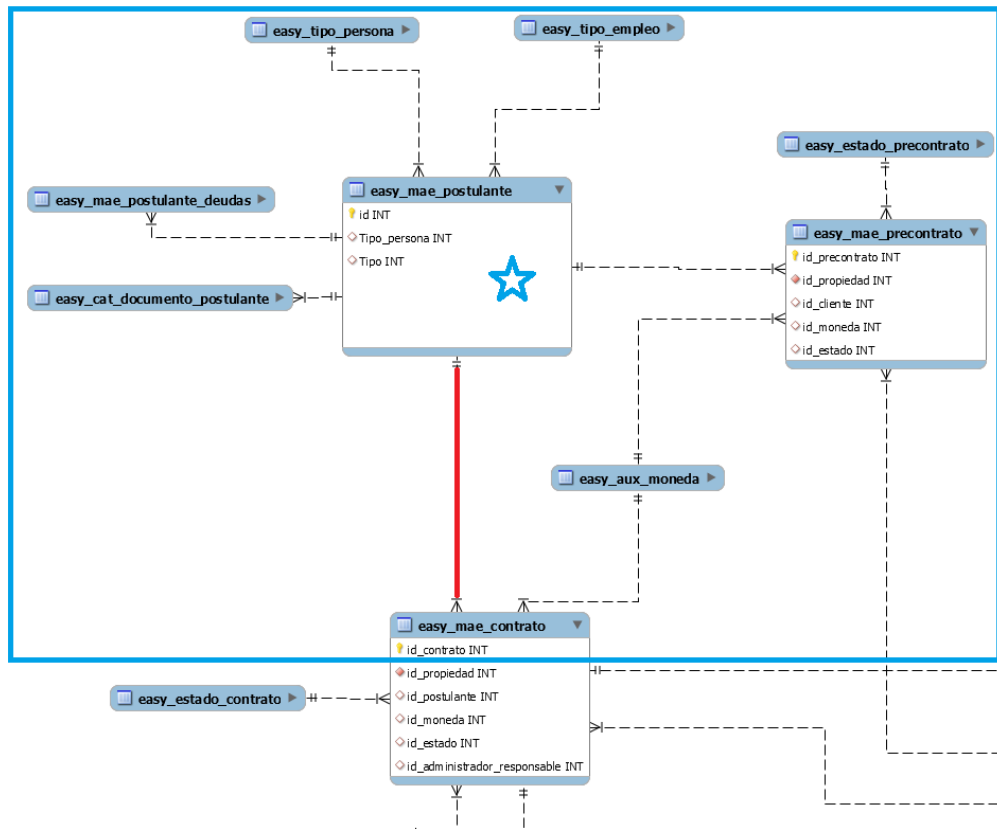


Figura 23. Detalle de la macro entidad "Postulante" donde se reflejan en rojo las relaciones principales y las otras relaciones que la complementan.

Fuente: elaboración propia

En la figura 23, se presenta en detalle la macro entidad "Postulante" y su tabla principal, resaltada con una estrella de color azul. La relación marcada en rojo representa la conexión que esta entidad tiene con la tabla principal de la macro entidad "Contrato". Esta

relación desempeña un papel fundamental al generar los contratos de arriendo y mantener un historial del comportamiento de un cliente en arriendos anteriores.

La interconexión entre estas dos macro entidades, junto con las relaciones establecidas con las macro entidades de “Gastos” e “Ingresos”, desempeñan un rol fundamental en el mantenimiento de un historial del comportamiento de los clientes a lo largo de los arriendos previos. Esto proporciona la oportunidad de evaluar la confiabilidad de los clientes que están registrados en el sistema, evitando la repetición de problemas pasados y fomentando la creación de relaciones de confianza con los clientes. Al mantener un registro detallado de los arriendos anteriores, el sistema puede analizar el comportamiento financiero de los clientes, su puntualidad en los pagos y su cumplimiento de las obligaciones contractuales. Esta información es relevante al momento de tomar decisiones relacionadas con la duración de los contratos y la selección de nuevos arrendatarios.

3.5.9 Liquidación

La macro entidad "Liquidación" desempeña un papel fundamental en el sistema al mantener un resumen mensual de los gastos e ingresos generados durante el mes. Esto permite generar un control de caja para cada propietario, proporcionando una visión clara de las transacciones financieras asociadas a sus propiedades. La tabla principal de esta entidad cuenta con 14 atributos que permiten registrar los datos relevantes para resumir lo sucedido durante el mes.

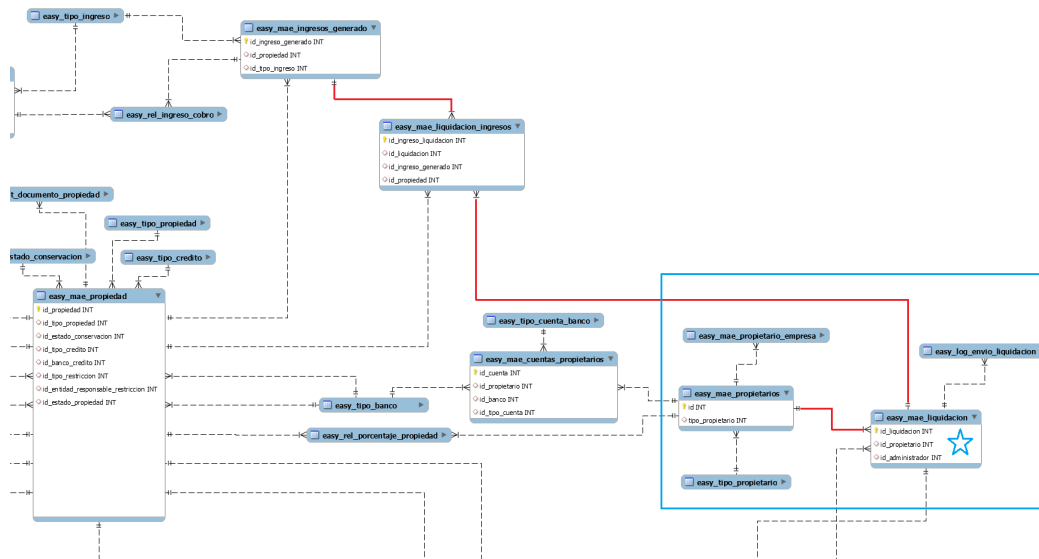


Figura 24. Primera parte del detalle de la macro entidad “Liquidación” donde se reflejan en rojo las relaciones principales y las otras relaciones que la complementan.

Fuente: elaboración propia

3.6 Diseño y usabilidad

El diseño de la aplicación web Easy Key se basa en la combinación de los principios de la psicología del color y las heurísticas de Nielsen, para que la interfaz de la aplicación cumpla con los estándares de usabilidad y facilidad de uso. Estos principios heurísticos, permiten diseñar interfaces intuitivas y eficientes, ayudando a minimizar los posibles traumas en los usuarios durante el uso de una aplicación. A continuación, se presentan los 10 principios que postula:

1. **Visibilidad y estado del sistema:** es relevante mantener al usuario informado sobre las operaciones que realiza el sistema, proporcionando una retroalimentación visual en tiempos razonables. Lo anterior tiene por objetivo evitar que el usuario tenga la sensación de que el sistema no esté funcionando con normalidad o que haya dejado de responder.
2. **Coincidencia entre el mundo real y el sistema:** al diseñar una aplicación, es fundamental considerar los elementos de la vida cotidiana y la experiencia del usuario con el objetivo de establecer una conexión efectiva entre los conceptos familiares para el usuario y los elementos visuales presentes en el sistema. Esta sincronización entre el mundo real y el sistema permite una comprensión más intuitiva y un aprendizaje fluido de la aplicación.
3. **Control y libertad del usuario:** cuando un usuario interactúa con algún sistema, existe la posibilidad de que se presenten errores ocasionados por la mala utilización de la aplicación. Por esta razón, es de vital importancia proporcionar al usuario una vía de escape que le permita corregir cualquier inconveniente de forma autónoma.
4. **Consistencia y estándares:** a lo largo del tiempo, los usuarios interactúan con diversos sistemas y aplicaciones. Debido a esto, se han establecido conceptos, iconos y estructuras que se utilizan para acciones específicas. En este sentido, este principio propone que las interfaces mantengan dichos elementos.
5. **Prevención de errores:** al momento de crear una interfaz de usuario, es posible que el usuario pueda equivocarse, es por esto que se deben generar interfaces que tengan medidas que eviten estos errores, proporcionando una retroalimentación clara y anticipada para guiar al usuario a generar las acciones de forma correcta.
6. **Reconocimiento para evitar el recuerdo:** las pantallas del sistema deben ayudar y guiar al usuario a realizar la acción que desea, evitando que el usuario tenga que memorizar los pasos a realizar para ejecutar la funcionalidad del sistema.

7. **Flexibilidad y eficiencia de uso:** la interfaz a crear debe permitir a los usuarios realizar acciones de forma rápida y eficiente, por medio de atajos que le ayuden a agilizar sus labores.
8. **Diseño estético y minimalista:** al momento de diseñar una aplicación se vuelve fundamental generar una interfaz limpia y estéticamente agradable que permita a los usuarios sentirse cómodos y atraídos por la aplicación, evitando elementos innecesarios que generan distracciones en los usuarios.
9. **Ayuda al usuario para reconocer, diagnosticar y recuperarse de los errores:** en caso de que no se haya podido evitar un error, es importante que la aplicación proporcione un mensaje de error claro y descriptivo que le permitan al usuario final corregir el problema de forma fácil y rápida.
10. **Ayuda y documentación:** las aplicaciones deben ofrecer recursos de ayuda y documentación que sean accesibles para los usuarios.

Como se mencionó anteriormente la aplicación "Easy Key" se diseñó bajo estos principios, con el propósito de crear una aplicación web que sea intuitiva y fácil de aprender. A continuación, se presentan las heurísticas aplicadas a lo largo de la plataforma "Easy Key" para lo cual se asoció a cada una un color específico para verificar su cumplimiento.

- Visibilidad y estado del sistema - Rosado
- Coincidencia entre mundo real y sistema - Verde
- Control y libertad del usuario - Celeste
- Consistencia y estándares - Amarillo
- Prevención de errores - Rojo
- Reconocimiento para evitar el recuerdo - Cafe
- Flexibilidad y eficiencia de uso - Gris
- Ayuda a los usuarios para reconocer, diagnosticar y recuperarse de los errores - Morado.

3.6.1 Colores

En la siguiente imagen se muestran los colores seleccionados para la aplicación Easy Key.

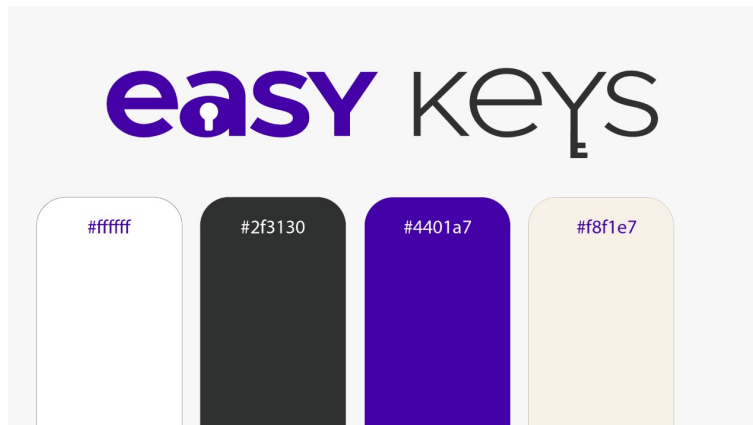


Figura 27. Paleta de colores principales de la aplicación EasyKey y su logo.
Fuente: elaboración propia

En la Figura 27 se pueden observar específicamente los colores seleccionados para la aplicación EasyKey, el primer color a destacar es el blanco, que según la psicología del color, sugiere elegancia y pureza, al igual que, el último color, que es un blanco más cálido, similar al crema, el cual transmite las mismas sensaciones de sofisticación y modernidad, estos tonos fueron elegidos para que los usuarios perciban el sistema como una aplicación actual y refinada, con el objetivo de fomentar el uso del desarrollo y de la tecnología en el rubro inmobiliario. Por otro lado, los dos colores que se encuentran al centro de la imagen, el gris y el morado, transmite seguridad, confianza y conocimiento, lo cual es fundamental en el ámbito inmobiliario, debido a que dentro de esta área se manejan grandes cifras de dinero y debido a eso, es crucial que la aplicación genere en el usuario la sensación de que los cálculos son precisos y confiables.

3.6.2 Diseño de las interfaces gráficas.

Para continuar con el diseño, la aplicación se basa en una sola interfaz gráfica, la cual es de uso exclusivo del administrador del sistema, el usuario tendrá la capacidad de realizar todas las tareas que están presentadas en las historias de usuario del administrador.

El diseño de la aplicación se centra en una pantalla inicial que presenta un menú lateral en el lado izquierdo de la pantalla. Este menú le permitirá al usuario navegar por la aplicación y acceder a sus tareas de forma rápida y sencilla, con la finalidad de reducir la cantidad de clics que el usuario debe realizar para realizar sus labores cumpliendo con la sexta y la octava heurística de Nielsen que hablan sobre priorizar la comprensión del sistema evitando la memorización y el enfoque de un diseño estético y minimalista. A continuación, se muestra el diseño del menú principal del panel administrativo de EasyKey.

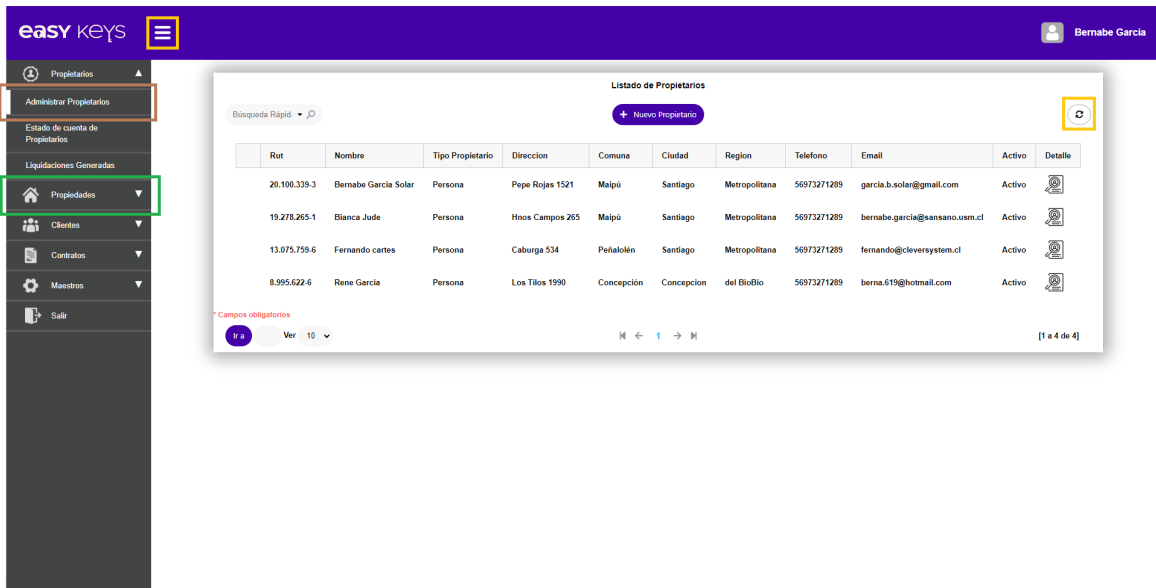


Figura 28. Diseño de menú principal panel de administración de Easy Key.

Fuente: elaboración propia

La Figura 28 muestra el menú lateral, la cual posee varias opciones disponibles para el administrador del sistema. En la imagen se muestran 4 heurísticas de Nielsen destacadas con diferentes colores, el color amarillo representa el cuarto principio que habla sobre la consistencia y estándares en el diseño, esta heurística se refleja por medio del ícono de apertura y cierra del menú que se encuentra en la parte superior izquierda y por el ícono que refresca la página que se encuentra en el formulario desplegado en la parte superior derecha, estos íconos han sido reconocidos por varios sistemas web, lo que permite que los usuarios relacionen el ícono a la funcionalidad explicada.

Lo que se encuentra destacado con color verde, indica el segundo principio de Nielsen, que explica la coincidencia entre el mundo real y el sistema, el cual se representa por medio de los íconos del menú, específicamente el ícono de la casa que simboliza las propiedades del área inmobiliaria. También se puede observar en el menú el ícono de un documento que hace referencia a los contratos y el símbolo de una tuerca para indicar la parte de los maestros. Por último, se destaca con color café el sexto principio, el cual postula que el usuario debe estar en conocimiento sobre dónde se encuentra durante el uso de la aplicación, esto se representa por medio de la diferenciación de la pestaña activa del menú por el color blanco al inicio del cuadro.

Continuando, en la figura 29 se muestran varios botones, incluido el botón de nuevo propietario, el cual conduce al usuario al siguiente formulario.

The image shows a web application interface for creating a new owner. The header is purple with the 'easy keys' logo and a user profile 'Bernabe Garcia'. A dark sidebar on the left contains navigation items: 'Propietarios', 'Administrar Propietarios', 'Estado de cuenta de Propietarios', 'Liquidaciones Generadas', 'Propiedades', 'Clientes', 'Contratos', 'Maestros', and 'Salir'. The main content area is titled 'Crear Propietario' and contains the following form elements: a dropdown for 'Tipo Propietario' (selected 'Persona'), a 'Rut' field with a red asterisk and a mask '## ### ####-#', a 'Nombre' field with a red asterisk, a 'Direccion' field, a 'Comuna' dropdown (selected 'Iquique'), an 'Email' field with a red asterisk, a 'Telefono' field with a red asterisk and a mask '+56 9 9999 9999', and an 'Activo' dropdown (selected 'Activo'). A red box at the bottom left indicates 'Campos obligatorios'. A 'Volver' button is in the top right, and an 'Agregar' button is at the bottom center.

Figura 29. Formulario de creación de propietarios de Easy Key.
Fuente: elaboración propia

Como se puede observar en la figura 29 en la parte superior derecha, se incluye un ícono para volver, el cual permite al usuario regresar a la lista de propietarios. Esta funcionalidad responde a la posibilidad de que el usuario no quiera crear un nuevo propietario y necesite regresar al listado previo. De esta manera, se satisface la tercera heurística de Nielsen, la cual se centra en el control y la libertad del usuario.

Además, el formulario destaca los campos obligatorios, los cuales se marcan con un asterisco. Si el usuario no llena estos campos, la creación del propietario no se llevará a cabo y se mostrará un error en pantalla. Estos campos son esenciales para el registro del propietario. Al demarcar claramente estos campos obligatorios en la interfaz, se cumple con la quinta heurística de Nielsen, que hace hincapié en la prevención de errores por parte del usuario.

Sin embargo, siempre se debe tener en cuenta que el usuario puede no estar atento y, como resultado, intente agregar un nuevo propietario sin completar los campos obligatorios. Para manejar esta eventualidad, se diseñó la siguiente pantalla.

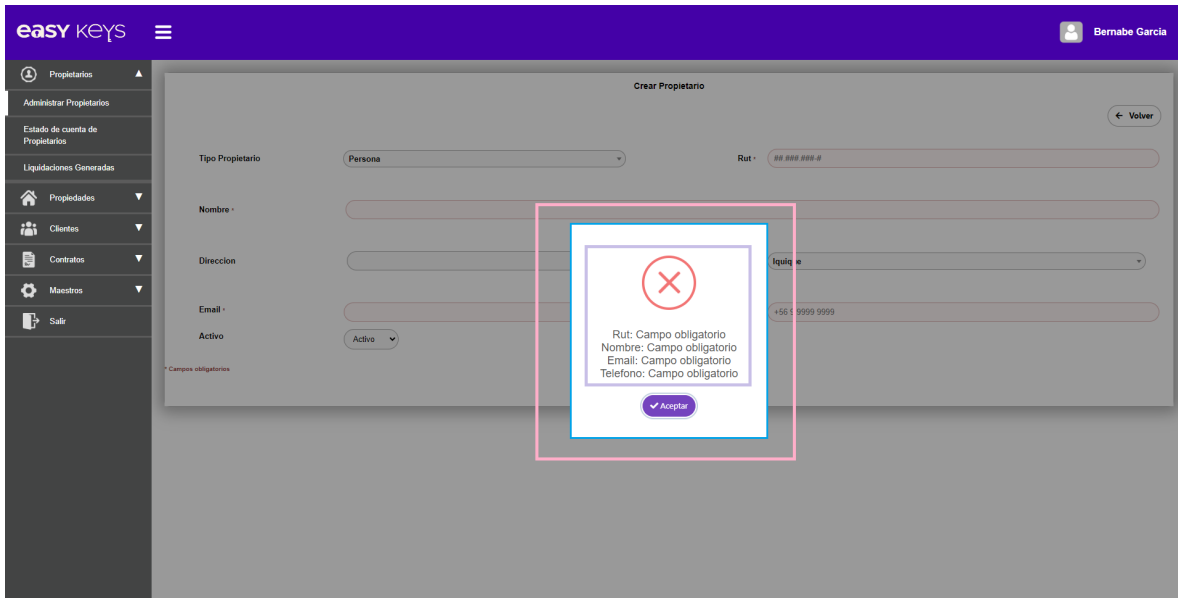


Figura 30. Error en el formulario de creación de propietarios de Easy Key.
Fuente: elaboración propia

En la figura 30, se puede observar claramente el error que cometió el usuario y que debe realizar para solucionarlo, cumpliendo con la primera y la novena heurística, que postulan la visibilidad y el estado del sistema, y la ayuda para el usuario que le permita reconocer y recuperarse de un error, estos principios están destacados en la imagen por medio de los colores rosado y morado respectivamente. Además, el usuario al identificar el error tiene la capacidad de poder solucionarlo volviendo al formulario y rellenando los campos no completados, cumpliendo así el tercer principio de Nielsen.

Luego de presionar aceptar, se puede observar la siguiente modificación de la pantalla de la figura 29.

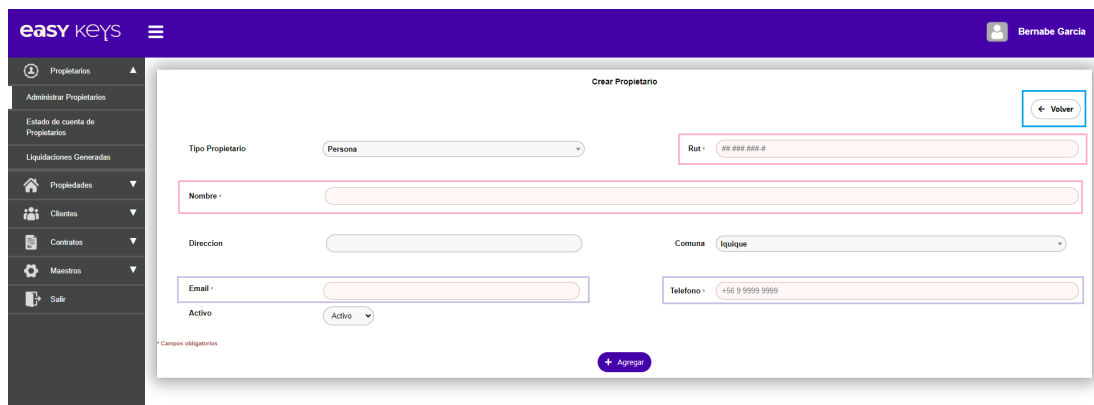


Figura 31. Formulario de creación de propietarios después del error.
Fuente: elaboración propia

Una vez generado el error, el sistema realiza un cambio de color en los campos obligatorios y que no fueron rellenos por el usuario, lo que permite cumplir nuevamente con la heurística 1 y 9, que permiten visualizar el estado del sistema y ayudar al usuario a cumplir la labor a ejecutar.

Luego de rellenar los campos solicitados y agregar al funcionario, se presenta un mensaje que le permite al usuario saber que su registro fue guardado correctamente.

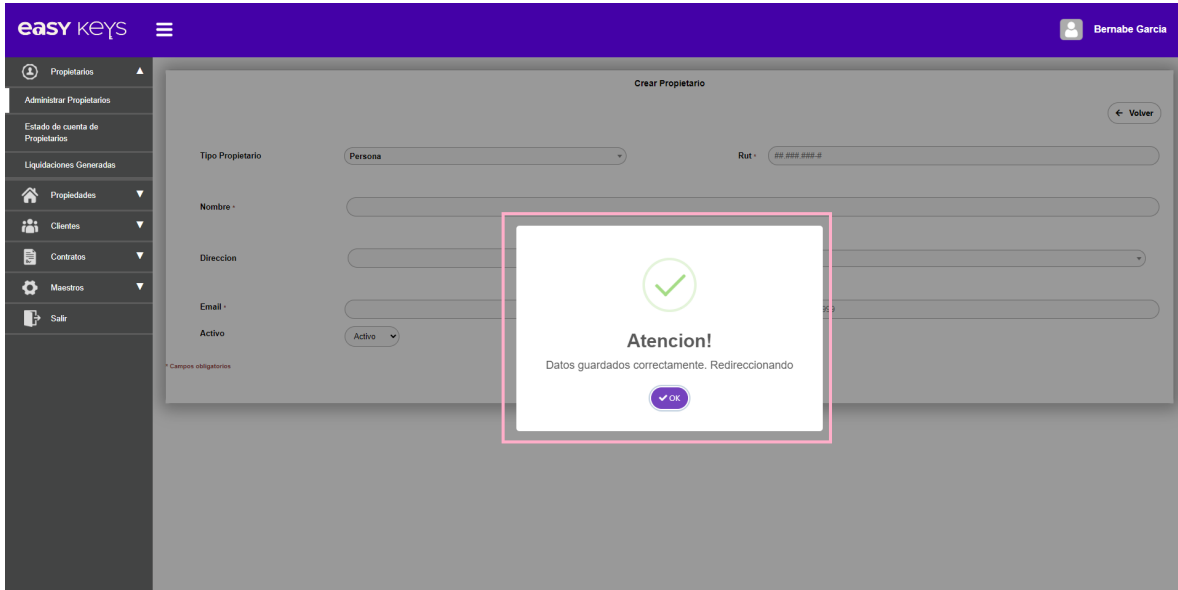


Figura 32. Proceso de creación de propietario exitoso.
Fuente: elaboración propia

Una vez realizado el ingreso del propietario, el sistema redirige al detalle del propietario creado, para que el usuario pueda revisar los datos guardados y pueda complementar con la información que falte por rellenar.

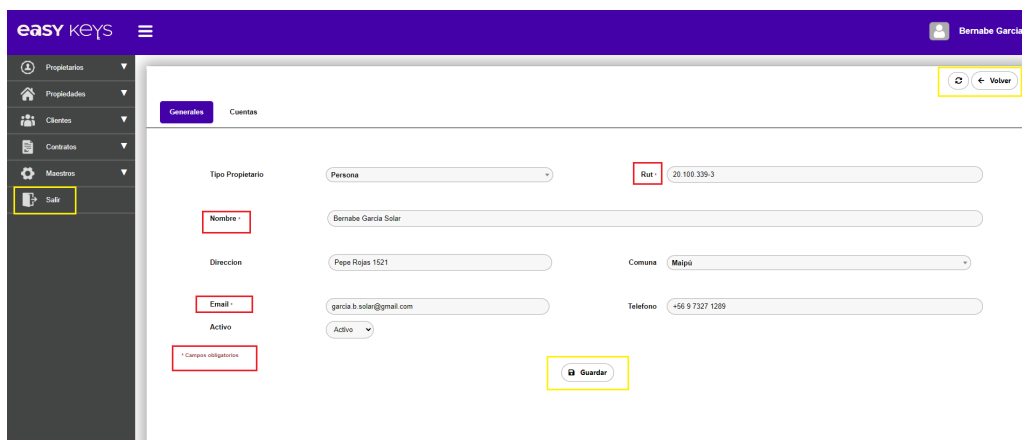


Figura 33. Pantalla del detalle de un propietario.
Fuente: elaboración propia

En la figura 33, se muestra la pantalla de detalle del propietario, la cual permite rellenar y modificar datos relevantes del mismo, en la imagen se destaca con color amarillo algunos íconos que ayudan a cumplir la cuarta heurística, y en color rojo aspectos que permiten cumplir la quinta heurística de Nielsen, la cual está relacionada con la prevención de riesgo. Por otro lado en la parte superior de algunos formularios, como es el formulario de la figura 33, se observan unas pestañas que permiten al usuario navegar por la opción seleccionada y ver información relacionada a la labor elegida, lo que ayuda a disminuir la cantidad de datos mostrados en pantalla, entregando una interfaz simple y ordenada.

Retomando la figura 28, en la lista de propietarios, están los íconos que representan los detalles individuales de cada propietario. Estos íconos tienen la finalidad de guiar al usuario al detalle del propietario seleccionado, y se muestran al igual que la figura 34, lo cual ayuda a no tener que reconocer dos pantallas distintas sino que por medio de una se pueda entender la mayor parte de la labor relacionada al CRU de Propietarios.

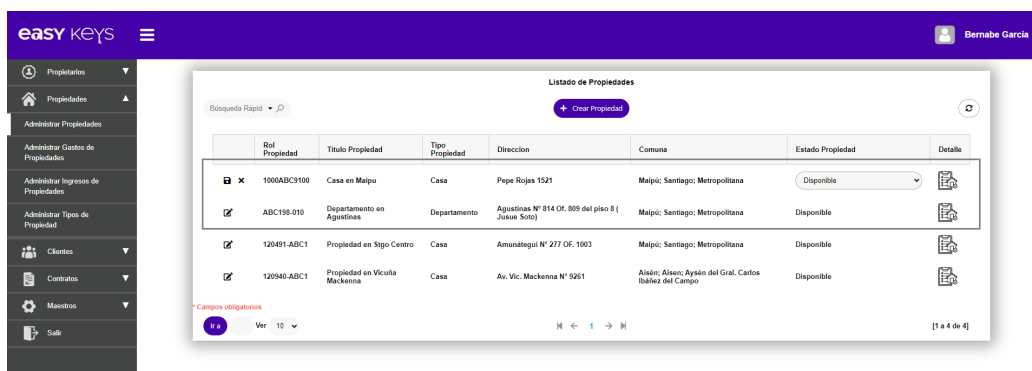


Figura 34. Pantalla del detalle de un propietario.
Fuente: elaboración propia

Por otro lado se tiene la figura 34, que permite modificar algunos datos específicos de una entidad, como en el caso de las propiedades, que por medio del ícono de edición que se encuentra en el lado izquierdo de cada fila, se puede editar de forma rápida el estado de la propiedad, evitando entrar al detalle de la propiedad y revisar datos que no son relevantes para la función de editar el estado de una propiedad, lo anterior, permite cumplir con el séptimo principio de Nielsen, el cual se relaciona con la flexibilidad y la eficiencia de uso en funcionalidades que se usan frecuentemente en una aplicación.

En general el diseño del área de administración se basa en las estructuras presentadas, un listado que puede ser de propiedades, gastos, ingresos, entre otros. Una pantalla que detalla los datos de una entidad específica y que además permite editarlos. La presencia de diferentes botones, los cuales tienen funcionalidades asociadas y los mensajes de aprobación o de falla que indican el estado del sistema.

La simplicidad del diseño de las interfaces expuestas, las cuales se enfocan en las funcionalidades del sistema, y que por medio de la selección de colores que según la psicología del color, entregan sensaciones de elegancia y seguridad, permiten al software desarrollado cumplir con la octava heurística de Nielsen, que habla sobre enfocar los esfuerzos en crear una aplicación que sea atractiva, evitando elementos innecesarios que provoquen la distracción de los usuarios finales.

3.6.3 Evaluación de la usabilidad de un software

Como se explicó en el subcapítulo anterior sobre el diseño, para la creación de EasyKey se siguió un enfoque basado en las heurísticas de Nielsen y la psicología del color con el objetivo de crear una plataforma de fácil uso e intuitiva, sin embargo existe una diferencia entre la evaluación teórica de un sistema y la realidad del uso, es por ello que a través del tiempo se han generado varias formas de evaluar la usabilidad de un sistema, esto se realiza por medio de los testing de usuarios y encuestas de usabilidad.

Los testing de usuarios son pruebas que se realizan a un sistema para comprender cómo los usuarios reales interactúan con él. Estas pruebas involucran a personas que representan al público objetivo del sistema. A los usuarios se les pide que realicen tareas específicas en el sistema mientras que los observadores registran su comportamiento. Esto proporciona información valiosa sobre las áreas donde los usuarios pueden tener dificultades, cómo interpretan las instrucciones y cómo navegan por la interfaz. Algunos tipos de pruebas son:

- **Pruebas Unitarias:** estas pruebas se centran en verificar que las partes más pequeñas y fundamentales de un sistema funcionen correctamente de manera aislada. Se prueban unidades individuales de código para asegurarse de que produzcan los resultados esperados.
- **Pruebas de Integración:** las pruebas de integración verifican que las diferentes unidades o componentes de software se combinen y funcionen en conjunto sin

problemas. Se aseguran de que las interacciones entre los componentes sean coherentes y que no haya conflictos.

- **Pruebas Funcionales:** estas pruebas evalúan si el software realiza correctamente las funciones para las que fue diseñado. Se centran en comprobar que las características y la lógica del sistema cumplan con los requisitos especificados.
- **Pruebas de Extremo a Extremo:** estas pruebas evalúan todo el flujo de trabajo de una aplicación, desde el inicio hasta el final. Se enfocan en las interacciones entre diferentes componentes y sistemas, asegurando que todo el proceso funcione como un todo integrado.
- **Pruebas de Aceptación:** las pruebas de aceptación se realizan para verificar si el sistema cumple con los requisitos del cliente y si está listo para ser entregado y utilizado. Pueden incluir pruebas de usuario final y pruebas de aceptación formal.
- **Pruebas de Rendimiento:** estas pruebas evalúan cómo el sistema se comporta bajo diferentes condiciones de carga, como el número de usuarios simultáneos, la cantidad de datos, etc. Se busca identificar cuellos de botella y problemas de rendimiento.
- **Pruebas de Humo:** las pruebas de humo son pruebas iniciales para verificar si una nueva versión del software está lo suficientemente estable como para realizar pruebas más exhaustivas. Se ejecutan pruebas básicas para detectar problemas graves antes de realizar pruebas más detalladas.

Además de los testing de usuarios, las encuestas de usabilidad también desempeñan un papel crucial. Estas encuestas recopilan la opinión y la percepción de los usuarios sobre la experiencia de uso del sistema. Los usuarios proporcionan retroalimentación en forma de comentarios y calificaciones sobre la facilidad de uso, la claridad de las instrucciones y la satisfacción general con la plataforma. A continuación, se presentan los tipos de encuestas de usabilidad más conocidos.

- **QUIS (Questionnaire for User Interface Satisfaction):** el cuestionario QUIS se utiliza para evaluar la satisfacción de los usuarios con la interfaz de usuario de un producto o sistema. Contiene una serie de preguntas que abordan diferentes aspectos de la experiencia del usuario, como la eficiencia, la facilidad de uso y la satisfacción general.
- **PUEU (Perceived Usefulness and Ease of Use):** el cuestionario PUEU se enfoca en medir la percepción de los usuarios sobre la utilidad percibida y la facilidad de uso de un sistema. Estas dos dimensiones son fundamentales para comprender la aceptación y la experiencia del usuario con respecto a una tecnología.
- **CSUQ (Computer System Usability Questionnaire):** el cuestionario CSUQ se utiliza para evaluar la usabilidad de sistemas informáticos. Contiene preguntas que abordan la eficiencia, la facilidad de aprendizaje y otros aspectos de la experiencia del usuario.
- **ASQ (After-Scenario Questionnaire):** el cuestionario ASQ se utiliza para evaluar la usabilidad después de que los usuarios hayan completado escenarios específicos de uso. Se centra en la satisfacción del usuario después de una interacción con el sistema.

- **PUTQ (Purdue Usability Testing Questionnaire):** es un cuestionario de prueba de usabilidad que aborda una amplia variedad de aspectos relacionados con la experiencia del usuario y la usabilidad de un sistema.

Para la evaluación del sistema EasyKey se eligieron 10 personas, de las cuales 6 personas son personas naturales y 4 son personas que tienen experiencia en el área inmobiliaria, a los testadores se les solicitó realizar 5 labores dentro del sistema, estas pruebas unitarias fueron elegidas debido a que son tareas que el usuario final va a realizar con frecuencia y son las primeras labores que va a desarrollar el administrador al momento de administrar las propiedades, además son labores que alimentan al sistema y le permiten generar informes valiosos para el administrador. Las labores por realizar son:

1. Crear un Propietario.

Se espera que el usuario pueda registrar los principales datos del propietario, identificar en qué momento el registro se realizó sin problemas y pueda verificar el registro.

2. Crear una Propiedad.

Se espera que el usuario pueda crear una propiedad, pueda verificar la creación de la misma y que pueda comprender que el sistema permite complementar la información ingresada.

3. Agregar un propietario a la propiedad creada.

Se espera que el usuario identifique que la propiedad necesita un propietario y pueda agregar un propietario a la propiedad creada.

4. Agregar un gasto único a una propiedad en el mes actual.

Se espera que el usuario identifique que la propiedad generó un gasto en el mes actual y que pueda registrarlo en el sistema.

5. Agregar un ingreso a la propiedad creada.

Se espera que el usuario pueda ingresar el ingreso recibido por la propiedad y pueda registrarlo.

Luego de realizar las pruebas a los usuarios se les realizó un cuestionario que mezcla algunas preguntas del cuestionario CSUQ que evalúa la usabilidad del sistema y el aprendizaje del sistema, y preguntas de la encuesta ASQ que se enfoca en la sensación o satisfacción del usuario luego de tener una interacción con el sistema. Las sentencias de la encuesta se presentan a continuación.

N°	Sentencia
1	En general, estoy satisfecho con lo fácil que es usar este sistema.
2	Fue sencillo utilizar este sistema.
3	Me siento cómodo usando este sistema.
4	Fue fácil aprender a usar este sistema.

6	Cada vez que me equivoco al usar el sistema, me recupero fácil y rápidamente.
7	Es fácil encontrar la información que necesitaba.
8	La información proporcionada por el sistema es fácil de entender.
9	La información es efectiva para ayudarme a completar las tareas y escenarios.
10	La organización de la información en las pantallas del sistema es clara .
11	La interfaz de este sistema es agradable.
12	Me gusta usar la interfaz de este sistema.
13	En general, estoy satisfecho con la cantidad de tiempo que tomó completar las tareas en este escenario.
14	En general, estoy satisfecho con este sistema.

Tabla 2. Listado de sentencias obtenidas del cuestionario CSUQ y ASQ que evalúan la usabilidad de un sistema.

Fuente: elaboración propia

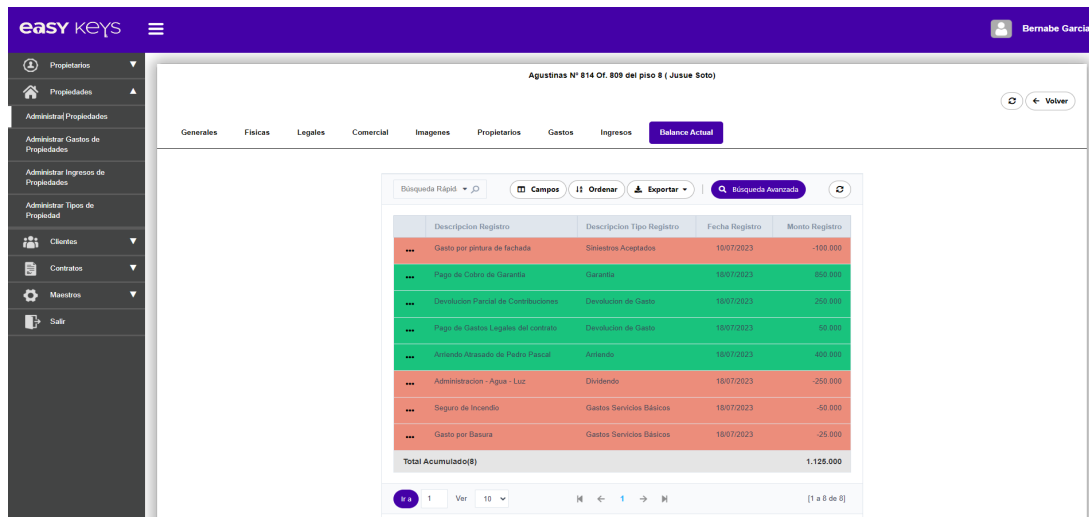
Cada sentencia se evalúa con un número del 1 al 7, donde el 1 significa que está muy en desacuerdo y el 7 quiere decir que el usuario está totalmente de acuerdo con la sentencia.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN.

Para validar la solución propuesta se deben recordar los objetivos de la aplicación EasyKey.

El objetivo general tiene el propósito de crear un software de fácil uso que ayude a los administradores de bienes raíces en las múltiples labores que conlleva el manejo de una propiedad, manteniendo un estado de caja controlado y ordenado de los inmuebles a su dominio. Los objetivos específicos por cumplir se basan en proponer una metodología que le permita al administrador tener un estado de cuenta actualizado y que por medio de la plataforma informe a los propietarios sobre sus propiedades. Lograr una buena gestión de los arriendos asociados a cada propiedad. Generar una aplicación que sea intuitiva y de fácil uso que cualquier tipo de usuario pueda utilizar por medio de la aplicación de metodologías y tecnologías de desarrollo de software que permitan la construcción de la plataforma ya diseñada.

Para cumplir con los objetivos propuestos, se creó la aplicación web Easy Key, que tiene diferentes funcionalidades que le permiten al usuario enfrentar varias actividades que tiene que realizar día a día, entre ellas están; registrar una nueva propiedad, registrar los gastos e ingresos de las propiedades, enviar las liquidaciones a los dueños de los inmuebles, entre otras. Todas estas funcionalidades le permiten al usuario tener todos los datos relacionados a la administración de propiedades centralizados y si estos son rellenados de forma oportuna dentro de la plataforma se logra generar un estado de cuenta como se puede observar en la siguiente imagen, este estado de cuenta considera los gastos, ingresos y la fecha de generación de cada uno de ellos, lo que permite tener una visión real y actual de las cuentas de la propiedad elegida.



Descripción Registro	Descripción Tipo Registro	Fecha Registro	Monto Registro
... Gasto por pintura de fachada	Sinistros Aceptados	10/07/2023	-100.000
... Pago de Cuota de Garantía	Garantía	18/07/2023	800.000
... Devolución Parcial de Contribuciones	Devolución de Gasto	18/07/2023	250.000
... Pago de Gastos Legales del contrato	Devolución de Gasto	18/07/2023	50.000
... Solución Arrendo de Pisos Pasivos	Arrendo	18/07/2023	400.000
... Administración - Agua - Luz	Dividendo	18/07/2023	-250.000
... Seguro de Incendio	Gastos Servicios Básicos	18/07/2023	-50.000
... Gasto por Basura	Gastos Servicios Básicos	18/07/2023	-25.000
Total Acumulado(\$)			1.125.000

Figura 35. Estado de cuenta de la propiedad Agustinas 813 Of. 809
Fuente: elaboración propia

PLATAFORMA PARA LA ADMINISTRACIÓN DE BIENES RAÍCES QUE BUSCA GENERAR LA TRANSFORMACIÓN QUE EL RUBRO INMOBILIARIO NECESITA

Por medio del control de gastos e ingresos de cada propiedad, se armó un estado global de cuenta que abarca a todas las propiedades de un propietario en específico, el cual contiene todas las transacciones del mes actual y es enviado al correo personal de cada uno, esto facilita la comunicación entre el administrador y los propietarios, y además le permite al administrador mantener informado a sus socios respecto a los márgenes de ganancia o pérdidas que generan sus propiedades administradas.

The screenshot shows a web interface for 'easy keys' with a sidebar menu on the left containing options like 'Propietarios', 'Administrar Propietarios', 'Estado de Cuenta de Propietarios', 'Liquidaciones Generadas', 'Propiedades', 'Clientes', 'Contratos', 'Masivos', and 'Salir'. The main content area displays a table titled 'Rosa Solar' with columns for 'Nº', 'Descripción Registro', 'Título Propiedad', 'Tipo Registro', 'Dirección', 'Monto', 'Porcentaje', 'Monto Propietario', and 'Fecha Registro'. The table lists various transactions such as 'Arrendo Atrasado de Pedro Pascal', 'Devolucion Parcial de Contribuciones', 'Devolucion de agua por corte', 'Pago de Gastos Legales del contrato', 'Pago de Cobro de Garantia', 'Gastos de Administracion', 'Seguro de Incendio', 'Dividendo Mensual', 'Contribuciones', 'Administracion - Gastos Comunes', 'Seguro contra incendios', and 'Pago de Gastos Legales'. A total row at the bottom shows 'Total Acumulado(13)' with a total amount of 1,912,000 and a total property amount of 631,000,000.

Figura 36. Estado de cuenta de la propietaria Rosa Solar.
Fuente: elaboración propia

Liquidación Arriendos		mes	JULIO	año	2023
Señor(a): ROSA SOLAR					
Arriendos Percibidos:					
Propiedad	Renta	Fecha Pago	%	Liquido	
Agustinas Nº 814 Of 809 del piso 8 (Jusue Soto) - Arriendo Atrasado de Pedro Pascal	400.000	10-07-2023	50	200.000	
Agustinas Nº 814 Of 809 del piso 8 (Jusue Soto) - Devolucion Parcial de Contribuciones	250.000	10-07-2023	50	125.000	
Alameda Nº 1845 Dpto. 401 - Devolucion de agua por corte	10.000	10-07-2023	50	5.000	
Agustinas Nº 814 Of 809 del piso 8 (Jusue Soto) - Pago de Gastos Legales del contrato	50.000	12-07-2023	50	25.000	
Av. Matta Nº 949 dp. 1006 bd 22-B - Pago de Gastos Legales	50.000	12-07-2023	100	50.000	
Agustinas Nº 814 Of 809 del piso 8 (Jusue Soto) - Pago de Cobro de Garantia	850.000	16-07-2023	50	425.000	
Av. Matta Nº 949 dp. 1006 bd 22-B - Pago de Garantia del Arriendo	550.000	17-07-2023	100	550.000	
TOTAL ARRIENDOS A PERCIBIR					1.380.000
GASTOS		MONTO	Fecha Ingreso	%	
Alameda Nº 1845 Dpto. 401 - Gastos de Administracion		200.000	20-07-2023	50	100.000
Alameda Nº 1845 Dpto. 401 - Seguro de Incendio		48.000	27-07-2023	50	24.000
Amunátegui Nº 277 OF. 1003 - Dividendo Mensual		400.000	27-07-2023	50	200.000
Amunátegui Nº 277 OF. 1003 - Contribuciones		150.000	27-07-2023	50	75.000
Av. Matta Nº 949 dp. 1006 bd 22-B - Administracion - Gastos Comunes		250.000	28-07-2023	100	250.000
Av. Matta Nº 949 dp. 1006 bd 22-B - Seguro contra incendios		100.000	30-07-2023	100	100.000
				TOTAL	749.000
Liquidación :					
Total Arriendos	1.380.000				
Gastos	749.000				
10% Administracion	138.000				
Saldo a favor	493.000				

Figura 37. Liquidación enviada a la propietaria Rosa Solar.
Fuente: elaboración propia

PLATAFORMA PARA LA ADMINISTRACIÓN DE BIENES RAÍCES QUE BUSCA GENERAR LA TRANSFORMACIÓN QUE EL RUBRO INMOBILIARIO NECESITA

Liquidación Arriendos mes de NOVIEMBRE de 2020				
Señor: CRISTIAN SUAZO				
Arriendos Percibidos:				
Propiedad	Renta	Fecha Pago	%	Liquido
Puchuncavi	0	-----	50	0
Eyzaguirre 229 -235 -	1.216.800	12-11-2020	50	608.400
TOTAL ARRIENDOS A PERCIBIR				\$ 608.400
GASTOS		MONTO	%	
Seguro de Incendio Propiedades	C: 7112	59.228	50	29.614
Puchuncaví - Cta. de luz		1.324	50	662
Puchuncaví - Cuidador		200.000	50	100.000
CBR - Copia con vigencia constructora Haltra		4.600	100	4.600
Puchuncaví - Set de fotocopias para estudio de		3.700	50	1.850
Puchuncaví - Correos de Chile envío documentos		4.580	50	2.290
			50	0
			50	0
Contribuciones: Cuarta cuota de 2020				
Eyzaguirre 229	3/3 primera cuota 2020	101.968	50	50.984
Eyzaguirre 235		74.433	50	37.217
Via F-170 5A-2	3/3 primera cuota 2020	239.170	50	119.585
Via F-170 5A-2 Contribuciones	C: 3-2020	728.275	50	364.138
			50	0
Via F-30 5A-1		28.859	50	14.430
TOTAL				\$ 725.369
Liquidación:			Abonos	
Total Arriendos	\$ 608.400			
Gastos	\$ 725.369			
10% administración	\$ 60.840			
Saldo	-\$ 177.809		#####	Saldo

Figura 38. Liquidación que envía la empresa Arrecife a sus socios.
Fuente. Elaboración empresa Arrecife

En la figura 36 se puede observar el estado de cuenta del mes de Julio de la propietaria Rosa Solar y en la figura 37, se muestra la liquidación enviada a la propietaria, considerando el valor que le corresponde al administrador de propiedades por sus labores de administración, esta misma al ser enviada al correo personal de la propietaria facilita la comunicación entre el administrador y los propietarios de cada propiedad, cumpliendo con el primer objetivo específico planteado. Para la generación del estado de cuenta se conversó con el encargado del envío de las liquidaciones en la empresa Arrecife y se acordó que el informe a enviar a los propietarios no debía sufrir muchas modificaciones, por lo cual para el diseño de la liquidación de la figura 37, se utilizó la liquidación tipo mostrada en la figura 38, esto se debe a que la liquidación ya formaba parte de su sello de empresa y que este informe ya se había modificado varias veces con anterioridad para que todos sus socios comprendieran cada parte del resumen mensual.

Por otro lado, se realizó el testing de usuarios y las encuestas de usabilidad en el marco de la evaluación de la usabilidad del sistema EasyKey, con estas herramientas se

obtuvieron datos significativos que ilustran la usabilidad y la satisfacción del usuario. Los resultados de estas evaluaciones proporcionan una comprensión detallada sobre cómo los usuarios interactúan con la plataforma y cómo perciben su experiencia de uso. A continuación, presentamos los resultados claves:

- **Testing de Usuarios**

Para evaluar el sistema se realizó un testing de software con un grupo de 10 individuos, que incluían tanto a usuarios naturales como a aquellos con experiencia en el ámbito inmobiliario. A continuación, se muestra la lista de usuarios, su edad y su manejo en tecnologías similares.

Nombre	Edad	Nivel conocimiento tecnológico	Trabajo relacionado al área Inmobiliaria
Maria Jose Correa	27	Básico	No
Constanza Correa	27	Básico	No
Bianca Jude	27	Medio	No
Fabio Pazos	24	Avanzado	No
Jonathan Gallardo	24	Medio	No
Cristobal Jude	20	Básico	No
Maria Jose Balmaceda	38	Avanzado	Si
Renato Colvin	57	Medio	Si
Patricio Colvin	53	Medio	Si
Sebastian Espinosa	35	Medio	Si

Tabla 3. Usuarios que participaron en el testing de la aplicación Easy Key
Fuente: elaboración propia

Como se puede observar en la tabla 3, el testing se realizó en personas con un rango etario amplio con el objetivo de que el testing pueda abarcar y conocer la opinión de diferentes generaciones, también se considera el conocimiento tecnológico del usuario, debido a que se busca recopilar los datos de un grupo amplio de personas con diferentes niveles de conocimiento en tecnologías similares con el objetivo de lograr que la herramienta sea fácil de usar para un amplio público, el nivel de conocimiento de cada usuario es subjetivo y es según su referencia personal sobre el uso de plataformas web parecidas.

Los resultados y conclusión de cada labor solicitada fue:

- **Labor “Crear un Propietario”**

Usuarios	Cantidad de usuarios que concluyeron la labor	Cantidad de usuarios que concluyeron la labor con ayuda	Tiempo Promedio
Sin Experiencia inmobiliaria	6	0	01:07
Con Experiencia inmobiliaria	4	0	01:00

Tabla 4. Tiempos promedios de la labor “Crear un propietario” y la cantidad de casos exitosos, separados por la experiencia inmobiliaria.

Fuente: elaboración propia.

Según los resultados expuestos en la tabla 4 los usuarios sin conocimiento inmobiliario se demoraron en promedio 1 minuto y 7 segundos en realizar la labor, en cambio, los que tenían experiencia inmobiliaria se demoraron 1 minuto y todos lograron concluir sin necesidad de la intervención del moderador, en esta labor no se encontró diferencias significativas al monitorear el comportamiento del usuario.

Nivel de experiencia en tecnologías	Cantidad de usuarios que concluyeron la labor	Cantidad de usuarios que concluyeron la labor con ayuda	Tiempo Promedio
Básico	3	0	01:35
Medio	5	0	00:50
Avanzado	2	0	00:48

Tabla 5. Tiempos promedios de la labor “Crear un propietario” y la cantidad de casos exitosos, separados por la experiencia en tecnologías similares.

Fuente: elaboración propia.

Según los resultados expuestos en la tabla 5, todos los usuarios lograron terminar la labor sin la intervención del moderador, los usuarios con nivel básico se demoraron en promedio 1 minuto y 35 segundos en realizar la labor, los que tenían un nivel medio, marcaron un

tiempo promedio de 52 segundo y las personas con un nivel avanzado se demoraron 48 segundos, como se puede observar existe una pequeña mejoría en los tiempos según el nivel de experiencia en tecnologías similares.

- **Labor “Crear una Propiedad”**

Usuarios	Cantidad de usuarios que concluyeron la labor	Cantidad de usuarios que concluyeron la labor con ayuda	Tiempo Promedio
Sin Experiencia inmobiliaria	6	0	01:59
Con Experiencia inmobiliaria	4	0	01:26

Tabla 6. Tiempos promedios de la labor “Crear una propiedad” y la cantidad de casos exitosos, separados por la experiencia inmobiliaria.

Fuente: elaboración propia.

Según los resultados expuestos en la tabla 6 los usuarios sin conocimiento inmobiliario se demoraron en promedio 1 minuto y 59 segundos en realizar la labor, en cambio, los que tenían experiencia inmobiliaria se demoraron 1 minuto y 26 segundos, y todos lograron concluir sin necesidad de la intervención del moderador, en esta labor si se encontró una diferencia que se debió a que habían algunos conceptos que los usuarios sin experiencia inmobiliaria no sabían el significado, lo que provocó que el moderador tuviera que explicar el concepto.

Nivel de experiencia en tecnologías	Cantidad de usuarios que concluyeron la labor	Cantidad de usuarios que concluyeron la labor con ayuda	Tiempo Promedio
Básico	3	0	02:27
Medio	5	0	01:33
Avanzado	2	0	01:07

Tabla 7. Tiempos promedios de la labor “Crear una propiedad” y la cantidad de casos exitosos, separados por la experiencia en tecnologías similares

Fuente: elaboración propia.

Según los resultados expuestos en la tabla 7, todos los usuarios lograron terminar la labor sin la intervención del moderador, los usuarios con nivel básico se demoraron en promedio 2 minutos y 27 segundos en realizar la labor, los que tenían un nivel medio, marcaron un tiempo promedio de 1 minuto y 33 segundos, y las personas con un nivel avanzado se demoraron 1 minuto y 7 segundos, al igual que la labor anterior, se observa una disminución en los tiempos según el nivel de experiencia en tecnologías.

- **Labor “Agregar un propietario a la propiedad creada”**

Usuarios	Cantidad de usuarios que concluyeron la labor	Cantidad de usuarios que concluyeron la labor con ayuda	Tiempo Promedio
Sin Experiencia inmobiliaria	3	3	01:49
Con Experiencia inmobiliaria	3	1	01:18

Tabla 8. Tiempos promedios de la labor “Agregar un propietario a la propiedad creada” y la cantidad de casos exitosos, separados por la experiencia inmobiliaria.

Fuente: elaboración propia

Según los resultados expuestos en la tabla 8 los usuarios sin conocimiento inmobiliario se demoraron en promedio 1 minuto y 49 segundos en realizar la labor, en cambio, los que tenían experiencia inmobiliaria se demoraron 1 minuto y 18 segundos, en esta labor hubieron algunos usuarios que necesitaron la ayuda del moderador para finalizar la tarea solicitada, a continuación se muestra la figura en donde se presentó el problema.

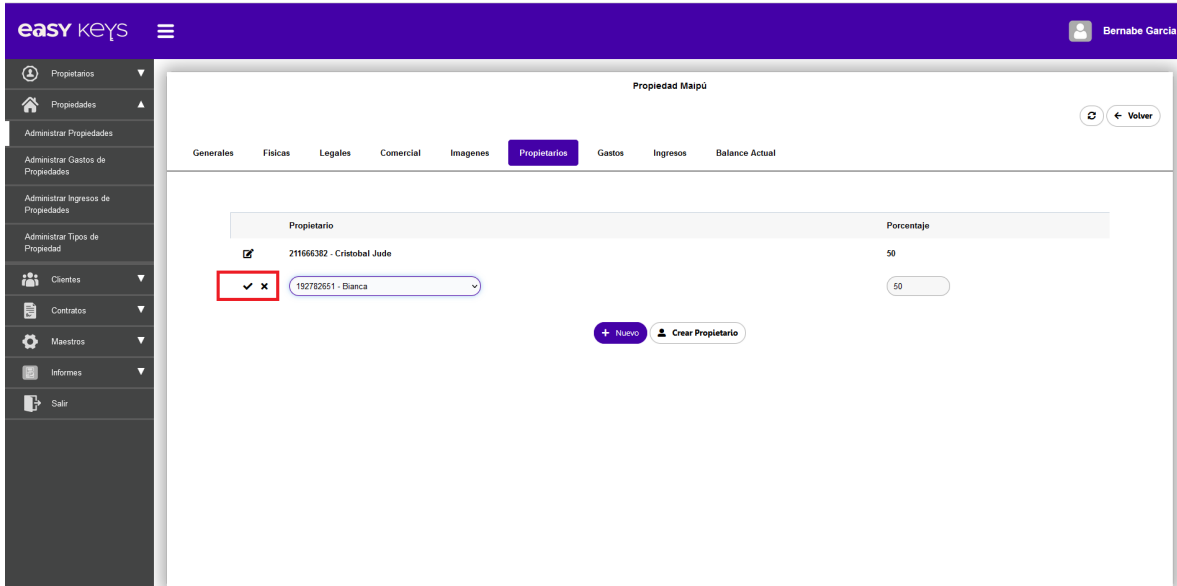


Figura 39. Pantalla donde se presenta el problema de la labor “Agregar un propietario a la propiedad creada”

Fuente: elaboración propia

Como se observa en la Figura 39 y según los comentarios de los usuarios que realizaron el testing, el botón “Crear Propietario” que tiene por función ser un acceso directo a crear un nuevo propietario, genera confusión, ya que los usuarios lo interpretan como un botón que genera el propietario de la propiedad, y que además el ticket no cumple la función de guiar al usuario a guardar el registro.

Nivel de experiencia en tecnologías	Cantidad de usuarios que concluyeron la labor	Cantidad de usuarios que concluyeron la labor con ayuda	Tiempo Promedio
Básico	1	2	02:06
Medio	3	2	01:33
Avanzado	2	0	00:49

Tabla 9. Tiempos promedios de la labor “Agregar un propietario a la propiedad creada” y la cantidad de casos exitosos, separados por la experiencia en tecnologías similares

Fuente: elaboración propia

Siguiendo adelante, los resultados presentados en la tabla 9 ofrecen una visión detallada de los tiempos promedio empleados por los usuarios en diferentes niveles de habilidad. Los usuarios con un nivel básico invirtieron un promedio de 2 minutos y 6 segundos para

completar la tarea; aquellos con un nivel medio registraron un tiempo promedio de 1 minuto y 33 segundos; mientras que los usuarios de nivel avanzado finalizaron la labor en tan solo 49 segundos. Como se destaca en la tabla 9, los usuarios que buscaron asistencia fueron aquellos que se encuentran en las categorías de nivel básico y medio. Sin embargo, resulta interesante observar que los usuarios de nivel avanzado, con experiencia en plataformas similares, lograron ejecutar la tarea sin dificultades. Este fenómeno insinúa que los usuarios de nivel básico y medio podrían no estar familiarizados con sistemas que empleen "tickets" como un ícono que genere la labor de guardar un registro. Esto, a su vez, entra en contraposición con la cuarta heurística de Nielsen, que está basada en el cumplimiento de estándares y mantener la consistencia del sistema. En este caso, se introdujo un elemento no universal para todos los tipos de usuarios. Es por esto que para asegurar la comprensión universal de la funcionalidad, podría ser necesario considerar la modificación del ícono para que se asemeje más a la acción, o incluso la sustitución por un botón explícito diga "guardar registro".

- **Labor "Agregar un gasto único a una propiedad en el mes actual"**

Usuarios	Cantidad de usuarios que concluyeron la labor	Cantidad de usuarios que concluyeron la labor con ayuda	Tiempo Promedio
Sin Experiencia inmobiliaria	1	5	02:04
Con Experiencia inmobiliaria	0	4	01:40

Tabla 10. Tiempos promedios de la labor "Agregar un gasto único a una propiedad en el mes actual" y la cantidad de casos exitosos, separados por la experiencia inmobiliaria.

Fuente: elaboración propia

Nivel de experiencia en tecnologías	Cantidad de usuarios que concluyeron la labor	Cantidad de usuarios que concluyeron la labor con ayuda	Tiempo Promedio
Básico	1	2	02:31
Medio	0	5	01:40
Avanzado	0	2	01:30

Tabla 11. Tiempos promedios de la labor “Agregar un gasto único a una propiedad en el mes actual” y la cantidad de casos exitosos, separados por la experiencia en tecnologías similares

Fuente: elaboración propia

Como se puede observar en la tabla 10 y 11 la mayoría de los usuarios necesitaron la intervención de un moderador para ejecutar la labor.

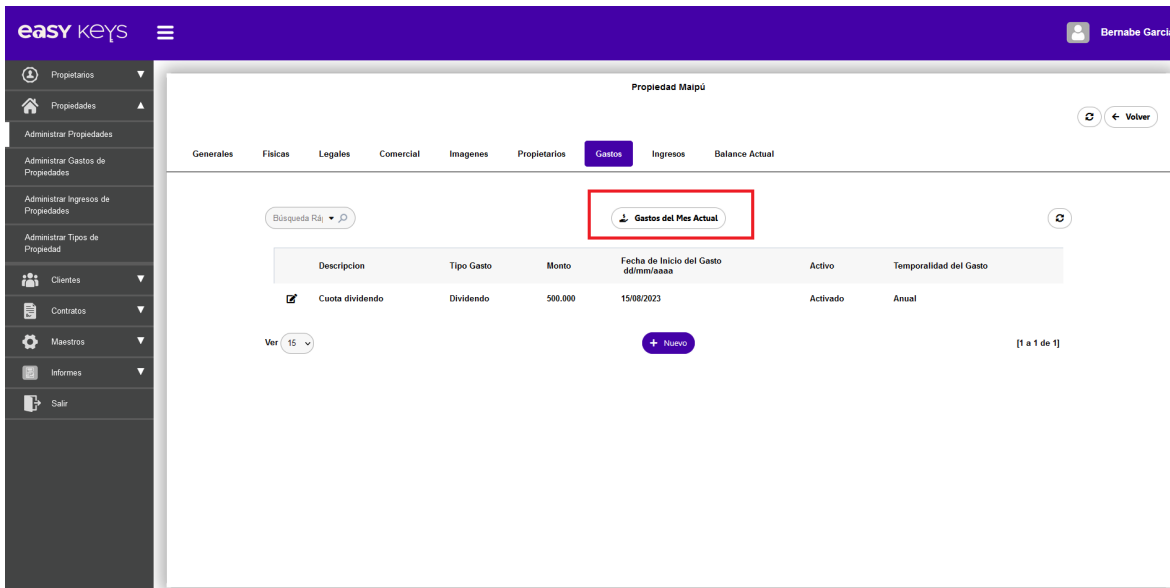


Figura 40. Pantalla donde se presenta el problema de la labor “Agregar un gasto único a una propiedad en el mes actual”

Fuente: elaboración propia

Según los comentarios recopilados de los usuarios, el problema principal se centró en el "botón" resaltado en rojo en la Figura 40. Este botón tenía la función de redireccionar al usuario a agregar un gasto único correspondiente al mes actual, la cual era la labor para realizar. Sin embargo, los usuarios interpretaron erróneamente este botón como un título del formulario en el que estaban ingresando. Como resultado, los usuarios generaron el

gasto en el formulario de la pestaña, que estaba diseñado para crear un gasto recurrente en el tiempo en lugar de un gasto único.

Este escenario presenta varias problemáticas. En primer lugar, falta un título que aclare la función del formulario en cuestión. Además, el botón de redirección comparte el mismo color de fondo y su borde no se distingue claramente, lo que contribuye a que el usuario se confunda. Por último, la pestaña no proporciona suficiente información descriptiva para que los usuarios comprendan el propósito del formulario al que accedieron.

Para abordar estas problemáticas, es fundamental realizar modificaciones en el diseño del formulario. Se sugiere generar un título que aclare la función del formulario actual y permita a los usuarios identificar dónde se encuentran en ese momento. Además, es crucial replantear la ubicación del botón de redirección. Una alternativa sería incorporar este botón como parte de las pestañas de navegación del formulario principal, de manera que las tareas asociadas a cada pestaña sean independientes y más comprensibles.

Al realizar estos ajustes, se espera que los usuarios puedan diferenciar claramente entre las diferentes labores a realizar en la plataforma y que su interpretación sea más precisa, mejorando así la experiencia general de uso y la usabilidad del sistema.

- Labor “Agregar un ingreso a la propiedad creada”

Usuarios	Cantidad de usuarios que concluyeron la labor	Cantidad de usuarios que concluyeron la labor con ayuda	Tiempo Promedio
Sin Experiencia inmobiliaria	6	0	00:42
Con Experiencia inmobiliaria	4	0	00:42

Tabla 12. Tiempos promedios de la labor “Agregar un ingreso a la propiedad creada” y la cantidad de casos exitosos, separados por la experiencia inmobiliaria.

Fuente: elaboración propia

Luego de la labor de crear un gasto único, se solicitó agregar un ingreso a la propiedad generada, los usuarios lograron completar la labor sin ningún problema con tiempos ajustados a lo esperado, en este caso como se puede observar en la tabla 12, ambos tipos de usuario lograron la labor en el mismo tiempo promedio.

Nivel de experiencia en tecnologías	Cantidad de usuarios que concluyeron la labor	Cantidad de usuarios que concluyeron la labor con ayuda	Tiempo Promedio
Básico	3	0	00:51
Medio	5	0	00:40
Avanzado	2	0	00:33

Tabla 13: Tiempos promedios de la labor “Agregar un ingreso a la propiedad creada” y la cantidad de casos exitosos, separados por la experiencia en tecnologías similares
Fuente: elaboración propia

En un enfoque más detallado, se evaluaron los resultados en función al nivel de experiencia en tecnologías de los usuarios. La tabla 13 ofrece una visión desglosada de los tiempos promedios y los niveles de éxito para cada nivel de experiencia: Básico, Medio y Avanzado. En este análisis, se observa una pequeña diferencia en los tiempos según el nivel de experiencia. Esta tendencia se ha observado consistentemente en los diferentes estudios realizados en el marco del testing de la plataforma EasyKey. Estos hallazgos reflejan cómo el grado de experiencia en el manejo de tecnologías puede influir en la velocidad de interacción con el sistema. Sin embargo, es importante destacar que, independientemente del nivel de experiencia, la mayoría de los usuarios lograron cumplir las tareas solicitadas, lo que refuerza la conclusión de que la plataforma es accesible para usuarios de diversos perfiles.

- **Encuestas de Usabilidad**

Las encuestas de usabilidad complementan los testing de usuarios al proporcionar una perspectiva más subjetiva de la experiencia del usuario. Los usuarios evaluaron el sistema por medio de 14 afirmaciones en una escala del 1 al 7, donde 1 representa "muy en desacuerdo" y 7 "totalmente de acuerdo". Los resultados de estas encuestas que se exponen a continuación ofrecen un panorama general de la satisfacción y percepción del usuario en relación con el sistema EasyKey.

N°	Sentencia	Promedio sin experiencia inmobiliaria	Promedio con experiencia inmobiliaria	Promedio Global
1	En general, estoy satisfecho con lo fácil que es usar este sistema.	6,5	6,5	6,5
2	Fue sencillo utilizar este sistema.	6,5	6,3	6,4

3	Me siento cómodo usando este sistema.	7,0	6,5	6,8
4	Fue fácil aprender a usar este sistema.	6,5	6,8	6,6
6	Cada vez que me equivoco al usar el sistema, me recupero fácil y rápidamente.	6,7	6,5	6,6
7	Es fácil encontrar la información que necesitaba.	6,5	6,8	6,6
8	La información proporcionada por el sistema es fácil de entender.	6,5	6,5	6,5
9	La información es efectiva para ayudarme a completar las tareas y escenarios.	6,2	6,3	6,2
10	La organización de la información en las pantallas del sistema es clara.	6,7	6,5	6,6
11	La interfaz de este sistema es agradable.	6,8	7,0	6,9
12	Me gusta usar la interfaz de este sistema.	7,0	7,0	7,0
13	En general, estoy satisfecho con la cantidad de tiempo que tomó completar las tareas en este escenario.	6,8	6,5	6,7
14	En general, estoy satisfecho con este sistema.	6,5	6,5	6,5
-	Promedio General	6,6	6,6	6,6

Tabla 14: Resultados promedios de la encuesta generada por preguntas de la CSUQ y ASQ post testing de usuario.

Fuente: elaboración propia

Como se puede observar en la tabla 14, la sensación de los usuarios fue bastante positiva. La mayoría de las calificaciones promedio superan el umbral de 6.5, lo cual indica que los usuarios experimentaron un nivel de satisfacción notable después de interactuar con el sistema. Vale la pena destacar las sentencias 11 y 12, que se refieren a la interfaz y a la

impresión general que esta genera en los usuarios. Estos aspectos son especialmente relevantes, ya que están directamente relacionados con el enfoque aplicado mediante el uso de la psicología del color y la octava heurística de Nielsen. La tendencia positiva en estas calificaciones subraya cómo la elección cuidadosa de colores y el diseño de la interfaz han logrado conectar con los objetivos deseados y con las mejores prácticas de usabilidad, contribuyendo a una experiencia satisfactoria y agradable para los usuarios.

Otra sentencia que destacar es la número 9, que dicta sobre la información desplegada en pantalla y si esta ayuda a completar las tareas solicitadas, este punto es el más bajo de toda la encuesta con una nota promedio de 6.2, y toma relevancia ya que tiene una conexión con lo encontrado en el testing y los comentarios realizados por los usuarios, ya que existen elementos en la plataforma que no aportan o que generan confusión en el usuario como se observó en las figuras 39 y 40.

Por otro lado, se puede observar que no existe una gran diferencia entre los promedios generados por los usuarios con y sin experiencia inmobiliaria, lo que indica que la sensación de los usuarios fue pareja para todos los participantes de la prueba y no se reflejaron grandes diferencias entre ellos, incluso el promedio general fue el mismo para todos los tipos de usuario, lo cual es un aspecto positivo, ya que permite afirmar que la aplicación cumple con ser una plataforma sencilla y fácil de usar para una gran diversidad de usuarios.

Con respecto a los resultados de los testing de usuarios y lo concluido de las encuestas de usabilidad se puede concluir que EasyKey ha logrado su cuarto objetivo el cual es ser una plataforma intuitiva y de fácil uso que cualquier usuario independiente de su experiencia en el área inmobiliaria pueda usar.

Por otro lado, la inserción de la aplicación EasyKey le permitió a la empresa Arrecife dar un gran paso hacia la transformación digital, ya que logró solucionar varias problemáticas que presentó el cliente en un inicio, logrando mejorar los procesos de gestión de bienes raíces. A continuación, se explica la forma en la cual la aplicación abordó algunas de las problemáticas planteadas.

- **Paso de registros físicos a digitales:** La aplicación permite la digitalización de todos los comprobantes, contratos y documentos relevantes para las propiedades y los arriendos, logrando que toda la información se encuentre almacenada de manera segura y respaldada en un entorno digital, eliminando la dependencia de los registros físicos y los riesgos que conlleva tenerlos de esa forma.
- **Generación de un canal de información centralizado:** Debido a que todos los registros se encuentran dentro de la aplicación y la información de los arriendos están disponibles en el sistema, ya no se depende del personal a cargo, logrando que cualquier persona con acceso al sistema pueda acceder a la información necesaria para tomar una decisión y asumir la responsabilidad en caso de que la

persona a cargo no esté disponible.

- **Automatización de los estados de cuenta:** A través del manejo de los datos registrados en el sistema y el poder de cómputo que tiene la aplicación, se pueden generar los estados de cuenta de cada propietario, ahorrando tiempo y recursos en estas labores que toman tiempo y son difíciles de realizar.

Los cambios en la forma trabajar y almacenar la información ha generado un gran cambio en la empresa Arrecife, permitiendo que esta empresa de un paso hacia la modernización y la optimización de sus procesos, cumpliendo con el último objetivo expuesto sobre la transformación digital y la manera de administrar un bien raíz.

Por último la metodología XP y las diversas tecnologías empleadas para el desarrollo de la aplicación EasyKey fueron implementadas de forma exitosa, generando una solución robusta que satisface las necesidades del cliente, lo cual evidencia el cumplimiento del segundo objetivo específico de este documento.

CAPÍTULO 5: CONCLUSIONES.

La aplicación EasyKey se alinea con el objetivo de introducir y fomentar la transformación digital en el sector inmobiliario. Este sector ha estado históricamente restringido por prácticas manuales y analógicas, lo que puede hacer que las operaciones sean lentas y propensas a errores. Con EasyKey, se introduce una herramienta digital intuitiva que optimiza y agiliza los procesos inmobiliarios, mejorando la eficiencia y precisión de las operaciones.

Esta transformación digital no solo tiene el potencial de cambiar las prácticas de administración de propiedades sino también de enriquecer la experiencia de los arrendatarios. Permite tener un control más preciso y en tiempo real del estado de las propiedades, lo que facilita la toma de decisiones informadas.

De este modo, EasyKey no solo busca resolver los desafíos presentes en la administración de propiedades, sino también impulsar al sector inmobiliario hacia el futuro. Esta transformación digital promueve la eficiencia, la exactitud y, sobre todo, una mejor comunicación y transparencia entre los arrendatarios y los propietarios.

Además de los logros alcanzados en la transformación digital de la industria inmobiliaria, es importante resaltar la experiencia vivida durante el proceso de desarrollo de EasyKey. Para llevar a cabo este proyecto, se optó por la utilización de la metodología de Desarrollo Ágil Extreme Programming (XP), la cual ayudó bastante a gestionar el desarrollo, conocer el estado del sistema en cada fase en el que se encontraba y lograr comunicar los plazos de entrega al cliente oportunamente.

La metodología XP demostró su eficacia al ofrecer una estructura flexible y colaborativa que fomentó la comunicación estrecha y efectiva entre el equipo de desarrollo y el cliente. Sin embargo, es importante mencionar que, si bien la metodología XP resultó en una dinámica fluida, el éxito de la comunicación dependía en gran medida de la disposición y la disponibilidad del cliente. Debido a esto se dieron situaciones en donde la comunicación se vio afectada, lo que provocó que se presentarán obstáculos para mantener el ritmo y la dirección del desarrollo.

En la implementación de EasyKey, se incorporó una herramienta de programación llamada ScriptCase, que desempeñó un papel significativo en la creación y la personalización de la plataforma. ScriptCase simplificó el proceso de desarrollo al proporcionar una interfaz visual y funcionalidades predefinidas que aceleraron la creación de módulos de la aplicación. Esto permitió que los esfuerzos se centraran más en la lógica y la funcionalidad específica del sistema, reduciendo el tiempo dedicado a tareas de programación repetitivas.

Sin embargo, es importante mencionar que, a pesar de las ventajas que aportó ScriptCase, también se encontraron limitaciones en términos de personalización avanzada y adaptabilidad a ciertas características únicas de EasyKey. Estas limitaciones a veces requirieron la intervención de programación adicional para lograr la funcionalidad exacta requerida, lo que generó ciertos desafíos y prolongó los tiempos de desarrollo en algunas instancias.

Los resultados de los testing aplicados, como las pruebas de usabilidad permiten constatar que EasyKey ha logrado ser una plataforma intuitiva y de fácil uso que cualquier usuario independiente de su experiencia en el área inmobiliaria pueda usar. Sin embargo, es importante mencionar que la validación de una solución tecnológica no es un proceso que se concluya de forma definitiva. Es necesario seguir iterando y mejorando EasyKey en base a la retroalimentación de los usuarios y a las cambiantes demandas del sector inmobiliario. Como paso futuro, se propone la realización de estudios de usabilidad más extensos y la inclusión de nuevas funcionalidades que añadan aún más valor a la experiencia del usuario.

La creación de EasyKey representa un avance significativo en la simplificación y optimización de la administración de propiedades. Este proyecto no solo ha logrado alcanzar los objetivos propuestos, sino que también ha allanado el camino para futuras innovaciones en el sector inmobiliario.

REFERENCIAS BIBLIOGRÁFICAS

1. Admltsqmet. (2022, diciembre 12). Los 8 tipos de metodología de desarrollo de software. ITSQMET; INSTITUTO TECNOLÓGICO SUPERIOR QUITO METROPOLITANO. <https://itsqmet.edu.ec/los-8-tipos-de-metodologia-de-desarrollo-de-software/>
2. Arenzana, D. (2016, marzo 30). Principios de usabilidad web de Jacob Nielsen y el diseño UX. Semrush Blog; Semrush. <https://es.semrush.com/blog/usabilidad-web-principios-jakob-nielsen/>
3. Arquitectura de software: Definición, elementos y tipos. (2023, febrero 17). Canvia. <https://canvia.com/arquitectura-software/>
4. Asistente, P., Darío, G., Alberto, D., Gutiérrez, B., & Último Semestre, E. (s/f). Scientia Et Technica. Redalyc.org. Recuperado en Julio de 2023 <https://www.redalyc.org/pdf/849/84917316032.pdf>
5. Atlassian. (s/f). Los distintos tipos de pruebas en software. Atlassian. Recuperado en Agosto de 2023 <https://www.atlassian.com/es/continuous-delivery/software-testing/types-of-software-testing>
6. Bello, E. (2021). Descubre qué es el Extreme Programming y sus características. Thinking for Innovation. <https://www.iebschool.com/blog/que-es-el-xp-programming-agile-scrum/>
7. Berthollet, M. (2022, julio 28). Heurísticas de usabilidad para la interfaz de usuario de Nielsen. Bring Connections. <https://bringconnections.es/tipos-heuristicas-usabilidad-nielsen/>
8. Biblioteca del Congreso Nacional. (s/f). Biblioteca del Congreso Nacional. www.bcn.cl/leychile. Julio de 2023. <https://www.bcn.cl/leychile/navegar?idNorma=1143741>
9. Biblioteca del Congreso Nacional de Chile. (2014). Biblioteca del Congreso Nacional de Chile. <https://www.bcn.cl/leyfacil/recurso/demanda-por-deuda-de-arriendo>
10. Canelo, M. M. (2020, julio 8). 12 librerías JavaScript que deberías conocer. Profile Software Services. <https://profile.es/blog/librerias-javascript/>
11. Canive, T. (2020, mayo 27). Metodología XP o Programación Extrema. Gestor de proyectos online.

<https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp>

12. Castelán, J. (2022, junio 28). 10 ventajas y desventajas de PHP. Talently Blog. <https://talently.tech/blog/ventajas-de-php/>
13. EBF. (2019, septiembre 11). Ventajas y desventajas de las metodologías Agile (ágiles). EBF - El blog de los mejores máster en A Coruña. <https://ebf.com.es/blog/ventajas-y-desventajas-de-las-metodologias-agiles-y-su-aplicacion-en-el-trabajo/>
14. el lenguaje Transact- SQL, E. un S. de G. de B. de D. R. B. en, Iq, y. E. en, & de tener unas ventajas que a continuación se pueden describir., C. de P. a. D. de M. U. G. C. de D. de M. S. A. (s/f). Microsoft SQL Server. Uaem.mx. Recuperado en Junio de 2023
<http://www.gridmorelos.uaem.mx/~mcruz/cursos/miic/sql5.pdf>
15. El mejor time-to-market para Java - OpenXava. (s/f). Openxava.org. Recuperado agosto de 2023
<https://openxava.org/es/>
16. Estilo de arquitectura basada en eventos. (s/f). Microsoft.com. Recuperado el 14 de junio de 2023, de
<https://learn.microsoft.com/es-es/azure/architecture/guide/architecture-styles/event-driven>
17. García, F. (2023, abril 22). Si estás buscando una base de datos potente y escalable, PostgreSQL. Codigosql.top.
<https://codigosql.top/postgresql/ventajas-desventajas/>
18. Huet, P. (2022, agosto 24). Arquitectura de software: Qué es y qué tipos existen. Openwebinars.net.
<https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>
19. Ibacache, V. A. (2018a, noviembre 15). Todo sobre la Ley de Arriendo y el Juicio de Arrendamiento en Chile. Misabogados.com.
<https://www.misabogados.com/juicios-de-arrendamiento>
20. Ibacache, V. A. (2018b, diciembre 14). Guía de Compraventa de Inmuebles en Chile: Todo lo que necesitas saber. Misabogados.com.
<https://www.misabogados.com/guia-compraventa-inmuebles>
21. La baja, E. M. de A. R. en C. ha E. un I. C. en L. Ú. Q. A. (s/f). EL MERCADO DE ARRIENDO RESIDENCIAL. Bcentral.cl. Recuperado en julio de 2023.

<https://www.bcentral.cl/documents/33528/2150281/RecuadroI-IV1-ElMercado.pdf/bb634b38-7f4d-e6cb-92de-2f7b02200246?t=1579197883043>

22. Lacerda, C. (s/f). 7 Razones para utilizar una herramienta RAD. Scriptcase Blog - Development, Web Design, Sales and Digital Marketing. Recuperado en agosto de 2023.
<https://scriptcaseblog.net/es/development-es/7-razones-para-utilizar-una-herramienta-rad-2/>
23. Leasity - Dale play a tu propiedad. (s/f). Leasity.cl. Recuperado en junio de 2023
<https://www.leasity.cl/>
24. Los 7 requisitos para firmar contrato de arriendo notarial. (2021, junio 15). Firma Virtual.
<https://firmavirtual.legal/requisitos-firmar-contrato-de-arriendo-notarial/>
25. Ltda, F. (s/f). FairRent - Sistema de Administración de Arriendo de Propiedades - Software- Fairware Ltda. Fairware.cl. Recuperado en junio de 2023.
<https://www.fairware.cl/FairRent.asp>
26. Mas, D. F. (2021, julio 16). Se disparan los impagos de arriendo. Diario Financiero.
<https://dfmas.df.cl/df-mas/hablemos-de/se-disparan-los-impagos-de-arriendo>
27. MashaMSFT. (s/f). El registro de transacciones (SQL Server). Microsoft.com. Recuperado el 12 de junio de 2023, de
<https://learn.microsoft.com/es-es/sql/relational-databases/logs/the-transaction-log-sql-server?view=sql-server-ver16>
28. Melo, S. (2020, junio 5). Beneficios de cambiar el papel por una solución digital. DataScope.
<https://datascope.io/es/blog/beneficios-de-cambiar-el-papel-por-una-solucion-digital/>
29. MERCEDES GROUP. (s/f). Para dar pasos efectivos en la compra y venta de propiedades. MERCEDES GROUP. Recuperado en julio de 2023.
<http://www.lasmercedesgroup.cl/mercedes/voces-del-experto/para-dar-pasos-efectivos-en-la-compra-y-venta-de-propiedades>
30. Metodologías Ágiles vs Tradicionales. (2022, mayo 9). POSITIVO.
<https://www.positivo.pro/blog/metodologias-agiles-vs-tradicionales/>
31. Montoto, O. C. (s/f). Estándares formales de usabilidad y su aplicación práctica en una evaluación heurística. Blogspot.com. Recuperado 2023.
<https://olgacarreras.blogspot.com/2012/03/estandares-formales-de-usabilidad-y-s>

[u.html](#)

32. Muguira, A. (2016, julio 4). Usabilidad web: Un test que te dará grandes resultados. QuestionPro.
<https://www.questionpro.com/blog/es/usabilidad-web-test/>
33. Muñoz, E. A. M. (1681099201000). La importancia de la arquitectura de software: cómo los patrones de diseño pueden mejorar la calidad del software. LinkedIn.com.
<https://www.linkedin.com/pulse/la-importancia-de-arquitectura-software-c%C3%B3mo-los-del-manr%C3%ADquez-mu%C3%B1oz/?originalSubdomain=es>
34. News Center Microsoft Latinoamérica. (2021, marzo 12). A un año de la pandemia: Pymes chilenas aceleraron su transformación digital. News Center Latinoamérica.
<https://news.microsoft.com/es-xl/a-un-ano-de-la-pandemia-pymes-chilenas-aceleraron-su-transformacion-digital/>
35. Novoseltseva, E. (2020, enero 14). Arquitectura dirigida por eventos: ventajas y modelos. Apiumhub.
<https://apiumhub.com/es/tech-blog-barcelona/arquitectura-dirigida-por-eventos/>
36. Porcentaje de personas en teletrabajo disminuye a un tercio del empleo – Cámara de Comercio de Santiago – CCS. (s/f). Ccs.cl. Recuperado en julio de 2023.
<https://www.ccs.cl/2020/12/21/porcentaje-de-personas-en-teletrabajo-disminuye-a-un-tercio-del-empleo/>
37. PostgreSQL: el gestor de bases de datos a fondo. (s/f). IONOS Digital Guide. Recuperado en junio de 2023.
<https://www.ionos.es/digitalguide/servidores/know-how/postgresql/>
38. Powell, Z. (2022, abril 20). Node.js vs PHP: Una Comparación Frente a Frente. Kinsta®; Kinsta.
<https://kinsta.com/es/blog/node-js-vs-php/>
39. Property management software. (2016, febrero 5). Buildium; Buildium - Property Management Software.
<https://www.buildium.com/>
40. Pursell, S. (2023, abril 17). Pruebas de usabilidad: definición, tipos y ejemplos. Hubspot.es.
<https://blog.hubspot.es/website/pruebas-usabilidad>
41. Qué es MySQL Ventajas y desventajas. (2020, julio 6). Com.mx; Hosting OFERTA: Web Hosting México ▷ Aldeahost.

<https://aldeahost.com.mx/que-es-mysql-ventajas-y-desventajas/>

42. Qué es Outsystems y para qué se utiliza. (s/f). Qué es Outsystems y para qué se utiliza. Recuperado en agosto de 2023.
<https://www.incentro.com/es-ES/blog/que-es-outsystems-para-que-sirve>
43. Rentger. (s/f). Rentger.com. Recuperado en junio de 2023, de
<https://www.rentger.com/>
44. Scriptcase - Herramienta de desarrollo rápido PHP Web. (s/f). Scriptcase. Recuperado en agosto de 2023.
<https://www.scriptcase.net/es/>
45. Transformación digital. Qué es y su importancia y relación con los datos. (s/f). Powerdata.Es. Recuperado en agosto de 2023.
<https://www.powerdata.es/transformacion-digital>
46. Un 72% de las empresas planea mantener algún grado de teletrabajo. (2022, abril 19). El Mostrador.
<https://www.elmostrador.cl/agenda-pais/2022/04/19/un-72-de-las-empresas-plan-ee-mantener-algun-grado-de-teletrabajo/>
47. User interface usability evaluation with web-based questionnaires. (s/f). Garyperlman.com. Recuperado en agosto de 2023
<https://garyperlman.com/quest/index.html>
48. Ventajas y desventajas de Cassandra. (2023, abril 11). Pros y contras.
<https://www.proscont.com/ventajas-y-desventajas-de-cassandra/>
49. Ventajas y Desventajas de Python. (2021, septiembre 22). KeepCoding Bootcamps.
<https://keepcoding.io/blog/ventajas-y-desventajas-de-python/>
50. Walther. (2023, abril 25). Descubre las Ventajas y Desventajas de Node.js para Potenciar tu Sitio Web. Tutoriales Dongee.
<https://www.dongee.com/tutoriales/ventajas-y-desventajas-de-nodejs/>
51. WBS (Work Breakdown Structure), qué es y cómo se usa. (2018, marzo 13). BibLus.
<https://biblus.accasoftware.com/es/wbs-workbreakdownstructure-que-es-y-como-se-usa/>