



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

EXTENSIÓN DEL SISTEMA DE GESTIÓN DE TICKETS DE INMOBILIARIA MAGAL

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

SEBASTIÁN ALEXANDER MORA ARTIGAS

PROFESOR GUÍA:
SERGIO OCHOA DELORENZI

PROFESOR CO-GUÍA:
JAVIER ZAMBRA FERNÁNDEZ

MIEMBROS DE LA COMISIÓN:
JOCELYN SIMMONDS WAGEMANN
EDGARD PINEDA LEONE

SANTIAGO DE CHILE
2024

Resumen

Magal es una empresa inmobiliaria con más de 50 años de experiencia en el rubro, la cual lleva a cabo el proceso de post-venta/post-arriendo de propiedades. Este proceso se enfoca en el manejo y solución de problemas que puedan experimentar los clientes en sus viviendas. Actualmente Magal cuenta con un sistema CRM (*Customer Relationship Management*) que le permite tener control sobre los reportes o solicitudes de clientes, y programar visitas para revisar dichas incidencias.

Dicho CRM tiene deficiencias que pueden significar un problema para la empresa. Por ejemplo, en torno a la seguridad, los clientes no cuentan con un sistema de registro; es decir, no hay autenticación de quien ingresa un reclamo o solicitud, permitiendo así que cualquiera pueda reportar problemas en las propiedades de otras personas. Por otro lado, el software no permite a un usuario normal (empleado de Magal) agregar nuevos edificios a la base de datos. Debido a eso, la empresa debe solicitar a un usuario externo la carga de nuevos datos en la base de datos del CRM, lo cual es insostenible en el mediano y largo plazo. Por último, no existe un módulo que entregue estadísticas, que permita a Magal analizar los datos obtenidos, y de esa manera mejorar la toma de decisiones estratégicas.

Para resolver esas deficiencias, en este trabajo de memoria se desarrolló una solución sobre el CRM existente, con el objetivo de agregar tres módulos que resuelvan los problemas descritos. Para el diseño de estos módulos, se discutió con Magal el flujo esperado del funcionamiento de cada uno, y a partir de ello, se tomaron decisiones para desarrollarlos. Estos módulos consideran la estructura general de la plataforma y también las interfaces existentes, para que las nuevas interfaces no difieran mucho de las anteriores, y permitan así mantener el aprendizaje del uso del sistema que ya tienen los empleados de Magal.

Posterior al desarrollo de los módulos, se hizo una validación a nivel de usabilidad y utilidad percibida por los usuarios. En la evaluación participaron dos administradores (empleados de Magal) y dos clientes reales. Los resultados obtenidos muestran que el sistema es considerado usable y útil por estos usuarios, por lo que se puede sostener que los objetivos de la memoria fueron cumplidos.

Finalmente, este trabajo presenta las conclusiones sobre la labor realizada. Además, se indican aspectos de mejora del sistema, como parte del trabajo a futuro.

*A Daniel y Loreto, mis padres.
Por todo su apoyo y creer en mí.*

Agradecimientos

Me es difícil expresar en palabras todo lo que agradezco y le debo a mis padres por haberme dado las facilidades de estudiar, y por haberme apoyado y confiar en mí, al igual que mis hermanos Ignacio y Marx quienes siempre me dieron aliento y apoyo durante los difíciles años universitarios.

Quiero agradecer a mis amigos de Underdogs, quienes han estado ahí por muchos años, incluso desde antes de la universidad, siempre con los partidos, las juntas, las risas y el apoyo, tanto en los buenos como en los malos momentos. Me ayudaron mucho cuando sentía que no podía sacar la carrera adelante, y me dieron relajo mental en momentos de estrés. Tengo muchos recuerdos que me llevaré por muchos años y que jamás olvidaré, espero que nuestra amistad siga por siempre. Ahora después de mucho, puedo decirles “Sí se pudo, cabros”.

También quiero darle las gracias a mis amigos de VCentral (y Lau) quienes me demostraron que la distancia no importa cuando la amistad es genuina. En un período tan complejo como fue la pandemia e incluso después, me acompañaron con muchas risas y apoyo. Espero que sigamos como ahora de aquí en adelante.

Por último, agradezco enormemente a dos personas que gracias a ellas este trabajo puede existir. La primera me ofreció un tema de memoria cuando me encontraba en una búsqueda y sin saber qué hacer, y la segunda aceptó ser mi profesor guía a pesar de tener muchos estudiantes a su cargo, y me orientó y apoyó durante todo el proceso. Gonzalo Figueroa y Sergio Ochoa, les agradezco todo su apoyo a lo largo de este trabajo.

Tabla de Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Situación Previa	2
1.3. Objetivos de la Memoria	3
1.4. Resumen de la Solución Desarrollada	4
1.5. Estructura del documento	5
2. Análisis del Sistema Legado	6
2.1 Arquitectura del sistema	6
2.2. Descripción del Front-End	7
2.3. Descripción del Back-End	11
2.4. Descripción del Modelo de Datos	12
2.5. Estado del Arte	15
2.6. Investigación de Librerías	16
3. Concepción de la Solución	18
3.1. Principales Requisitos de la Solución	18
3.2. Diseño de la Planilla Excel	19
3.3. Diseño del Módulo de Carga de Datos	21
3.4. Diseño del Módulo de Estadísticas	22
3.4.1. Diseño de la Interfaz	22
3.4.2. Cálculo de los Datos	23
3.5. Diseño del Módulo de Registro de Usuarios	24
3.6. Proceso de Carga Masiva de Datos	26
3.7. Cambios al Modelo de Datos	28
3.8. Validación del Registro de Cliente	30
3.9. Resumen	31
4. Implementación de la Solución	32
4.1. Módulos de Carga de Datos	32

4.2. Módulo de Registro de Clientes	37
4.2.1 Interfaces del Usuario Cliente	37
4.2.2 Vistas de Administración	43
4.3. Panel de Estadísticas	49
4.4. Paso a producción.	51
4.5. Resumen	52
5. Evaluación de la Solución	53
5.1. Detalle de los Instrumentos de Evaluación	53
5.2. Proceso de Evaluación	54
5.3. Resultados Obtenidos	55
6. Conclusiones y Trabajo a Futuro	58
Bibliografía	60

Capítulo 1

Introducción

La industria inmobiliaria en Chile se vio afectada por la pandemia, provocando la disminución en las ventas de viviendas nuevas, especialmente en el año 2022, el cual tuvo un descenso de un 38% en el tercer trimestre. A causa de lo anterior, se espera que en el año 2023 haya un repunte en este sector, y que nuevos clientes adquieran propiedades. Por lo tanto, las empresas del rubro deben otorgar un servicio adecuado, para que los clientes se sientan a gusto y además puedan hacer saber de desperfectos en las viviendas compradas [1].

En este sentido, esta memoria se enmarca en el proceso de post venta o post-arriendo de la empresa Magal. Esta empresa cuenta con más de 50 años en el rubro inmobiliario, y más de 1.500.000 m² construidos que avalan la calidad de su trabajo. En las distintas etapas del proceso de venta o arriendo de un inmueble, Magal debe proveer el servicio de *post-venta o post-arriendo*. Este servicio es la parte de la atención al cliente, por lo que se diseñan estrategias y llevan a cabo las acciones para mejorar la experiencia de los clientes. Esta área también se encarga de mantener una relación duradera con los arrendatarios y compradores, por lo que es importante para Magal que este servicio sea eficiente y eficaz.

Magal ya cuenta con un sistema que maneja el servicio de post venta y post arriendo, basado en la recepción de tickets enviados por los clientes a través del sitio web, los cuales son revisados y aprobados según corresponda, para su posterior atención. Sin embargo, el sistema actual presenta diversas dificultades que se describen en la siguiente sección.

1.1. Motivación

Antiguamente Magal manejaba todas las operaciones de post venta de forma manual, y gracias al desarrollo de un sistema CRM (Customer Relationship Manager) se estableció una comunicación más fluida con el cliente, y se pudo mantener un historial de los eventos ocurridos en las distintas propiedades que maneja la empresa [2]. A través de su CRM, Magal puede manejar los reclamos (tickets) de cada cliente y agendar visitas en una interfaz que no presenta mayores dificultades, así como ver la información correspondiente de cada persona que reporta un problema con su vivienda. Cada incidencia reportada genera un ticket de atención para Magal, quien debe hacerse cargo de su tratamiento.

Todos los registros de citas, problemas reportados, y demoras, entre otros, solo son visibles uno a uno en el CRM. No existe un módulo que permita a Magal ver estadísticas de todas las situaciones relevantes que debe abordar (por ejemplo, los problemas reportados por los

clientes con más frecuencia), o descargar dicha información de forma directa para realizar reportes manuales.

Debido a que la empresa es una inmobiliaria activa, y que en el futuro construirá nuevos edificios, es necesario que esos nuevos proyectos de vivienda sean agregados al sistema de post-venta. De esa manera, los nuevos clientes podrán realizar sus reclamos directamente a través del sitio web, y no tener que interactuar con la empresa a través de llamadas telefónicas o correos electrónicos.

El sitio no cuenta con un método para agregar nuevos proyectos, lo cual debe hacerse de forma manual. Particularmente, una persona con los permisos adecuados debe conectarse al backend del CRM (aplicación Web) y realizar los cambios pertinentes en la base de datos. Estas prácticas no son viables de mantener en el mediano y largo plazo, pues además de engorrosas son una fuente de potenciales errores.

Por las razones mencionadas, además de otras deficiencias que se reportan en la siguiente sección, es que en este trabajo de memoria se abordó la extensión y mejora del CRM existente, conservando el estilo de interfaz al que los usuarios y clientes ya están acostumbrados.

1.2. Situación Previa

Como se mencionó en la sección previa, hasta antes del desarrollo de esta memoria el proceso de post venta de Magal se apoyaba en un sitio Web (CRM), al cual los clientes se conectaban y podían reportar problemas en sus viviendas. Por su parte, los trabajadores de Magal podían iniciar sesión en el sitio Web, y ver los tickets (órdenes de revisión o reparación) pendientes y completadas para cada inmueble.

El sitio web tenía diversas deficiencias para apoyar el proceso de post venta y post-arriendo de propiedades, así como también funcionalidades incompletas y otras a implementar. Esto le impedía a Magal brindar un mejor servicio a los clientes. Entre los problemas más importantes del sistema legado están los siguientes:

- *No se podía agregar nuevos edificios (proyectos) a la base de datos a través del sitio Web.* En el sitio no se contaba con la opción de subir un archivo, ya sea un excel u otro formato, que contenga la información de un nuevo edificio con sus viviendas y clientes. La carga inicial de los proyectos al sitio en producción, era hecha de forma manual por una persona externa a Magal. Esta no es una práctica que se pueda mantener en el tiempo, pues Magal no cuenta con un desarrollador que pueda realizar directamente dicha tarea. Por lo tanto, el poder contar con un módulo que le permita a Magal realizar esto dentro del sitio mismo, y con un formato específico para ello, eliminará la dependencia de personas externas y hará a las operaciones de post-venta/post-arriendo más fluidas.

- *No se podía obtener estadísticas en base a los datos almacenados.* Magal no contaba con la posibilidad de ver un resumen con estadísticas del proceso de post venta. A medida que la cantidad de reclamos crecía, la opción de revisar manualmente uno a uno se volvía básicamente imposible. Haciendo uso del sistema legado, no era posible saber en qué podía estar fallando en los edificios que construye la empresa. Por ejemplo, no sabían si los materiales o productos que compraban eran de baja calidad, pues no podían determinar la frecuencia con que se repetían los problemas del mismo tipo. Es decir, no se podía tomar decisiones estratégicas porque no se contaba con la información adecuada. El módulo desarrollado en esta memoria permite a Magal observar los datos recolectados a través del sitio Web. Esto sirve para apoyar la toma de decisiones sobre el cambio de proveedores, prevención de problemas que se reporten con frecuencia, y el abordaje de la construcción de los departamentos de manera diferente, entre otras.
- *No había control sobre la generación de tickets de atención.* Particularmente, no había control sobre las personas que generaban los tickets de atención. Esto significaba que cualquier persona podía simular ser un cliente de Magal, y generar tickets falsos. Eso disparaba acciones innecesarias por parte del personal de Magal, con el correspondiente consumo de tiempo y recursos. Por otra parte, Magal requería poder contar con un servicio donde el reporte de un problema no representara una tarea engorrosa para los clientes.

1.3. Objetivos de la Memoria

El objetivo general de este trabajo de memoria fue mejorar y extender el CRM de Magal, el cual está destinado a apoyar el servicio de post-venta/post-arriendo, con el fin de que dicho servicio sea más eficiente, efectivo y completo. El nuevo sistema debía quedar implantado correctamente como aplicación web, para así poderse usar tanto en computadores de escritorio, como en tablets y teléfonos móviles. A continuación se presentan los objetivos específicos que se derivan del objetivo general.

1. Diseñar una planilla en formato Excel que facilite la carga inicial de los nuevos proyectos, e implementar una pestaña visible en el actual CRM para que los usuarios administradores puedan realizar la carga masiva de datos de nuevos edificios.
2. Diseñar e implementar un módulo de estadísticas que entregue información de apoyo a la toma de decisiones, sobre (al menos) los siguientes puntos relevantes:
 - a. Tiempo promedio que demora resolver un problema.
 - b. Demora promedio en agendar una visita a la vivienda, desde que se recibe un ticket (reclamo).
 - c. Cantidad de veces que se reporta un mismo tipo de problema.
 - d. Tipos de departamentos que presentan más problemas.

3. Rediseñar la página de reporte de tickets (reclamos) para, entre otras cosas, solicitar inicio de sesión a los clientes antes de poder hacer uso del sistema.
4. Poner el sistema en producción, garantizando un buen nivel de usabilidad y utilidad, tanto para los clientes como para el personal de Magal.

1.4. Resumen de la Solución Desarrollada

La solución propuesta para resolver los problemas de la empresa, es una extensión al actual CRM de Magal, que es una aplicación web del tipo PSEE (Post Service Enhancing Experience). Los nuevos módulos y páginas agregadas conservaron la misma estética del sistema legado, para que los usuarios de Magal no presenten mayores dificultades al utilizar las nuevas funcionalidades.

Para la implementación de la extensión, primeramente se definieron historias de usuario, basadas en el flujo esperado por parte de un usuario, ya sea personal de Magal o del cliente que reporta una incidencia. A partir de las historias de usuario se hizo un diseño de las interfaces de los módulos y de las páginas a implementar, los cuales se validarán con los usuarios utilizando prototipos. La Figura 1.1 muestra los actores (tipos de usuario) que participan en este proceso, los componentes a través de los cuales acceden a los datos del servidor.



Figura 1.1: Escenario de trabajo de la nueva solución

Tomando en cuenta las interfaces y el modelo de datos del sistema, se realizó la conexión entre ambos componentes, siguiendo la misma lógica de la aplicación actual para obtener la funcionalidad esperada. Una vez listo esto, se hicieron pruebas de usabilidad y utilidad con diversos tipos de usuarios del sistema, utilizando instrumentos estandarizados (ya probados en la industria). La retroalimentación recibida fue utilizada para mejorar las interfaces y la funcionalidad de la aplicación.

Actualmente el sistema se encuentra en producción, y Magal cuenta con un software mejorado, con nuevas funciones y módulos que le permiten un mejor manejo de la información y apoyo a la toma de decisiones, así como también brindar un mejor servicio de post-venta / post-arriendo.

1.5. Estructura del documento

El resto del documento se estructura de la siguiente forma: en el Capítulo 2 se presenta un análisis de la plataforma y de la arquitectura del sistema intervenido. En el Capítulo 3 se presenta el diseño y desarrollo de la solución, para alcanzar los objetivos propuestos, considerando el estado actual del sistema. En el Capítulo 4 se describe la implementación de los nuevos servicios y módulos. El Capítulo 5 explica la validación realizada a la solución, a nivel de usabilidad y utilidad percibida por los usuarios. Finalmente, en el Capítulo 6 se presentan las conclusiones del trabajo realizado, y se plantean posibles mejoras al sistema (trabajo a futuro).

Capítulo 2

Análisis del Sistema Legado

El análisis del software legado se ha dividido en tres partes: arquitectura del sistema, front-end y back-end. El análisis de la arquitectura muestra la estructura general del sistema y las tecnologías utilizadas, las cuales se mantendrán en la actual extensión del mismo. El análisis del front-end considera el sitio Web como tal y sus componentes. El análisis del back-end incluye el modelo de datos, la arquitectura del sistema y la lógica de negocio asociada a los servicios ofrecidos por la plataforma. A continuación se describen estas tres partes.

2.1 Arquitectura del sistema

De acuerdo con lo estudiado y lo descrito en la memoria de Pablo Miranda [2], desarrollador del sistema legado, la arquitectura del mismo es híbrida. Particularmente, ésta combina una arquitectura cliente-servidor y una de tres capas. En la Figura 2.1 se muestra un diagrama de la arquitectura actual, además de las tecnologías que utiliza el sistema en cada una de las capas. A continuación se describirán brevemente las arquitecturas del sistema.

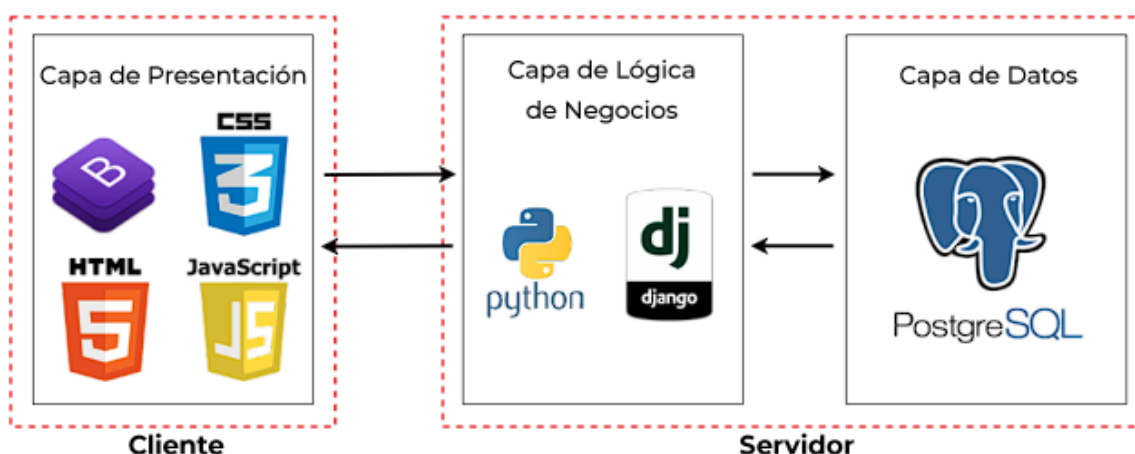


Figura 2.1: Arquitectura híbrida cliente-servidor y arquitectura de tres capas.

Arquitectura Cliente Servidor

Cuenta con dos componentes, el servidor, donde está gran parte de la lógica de negocio y almacenamiento de datos, y el cliente, que corresponde al código que se ejecuta en el equipo

del usuario. Ambos componentes se comunican a través del protocolo HTTP, dado que el sistema es una aplicación web.

Arquitectura de tres capas

Separa el software en tres capas: presentación, lógica de negocio y datos. Particularmente, el “servidor” descrito en el punto anterior contiene las dos últimas capas, tal como se puede ver en la Figura 2.1.

La capa de presentación consiste en la interfaz gráfica y componentes básicos de lógica, como validación de datos y emisión de alertas. Las tecnologías utilizadas en esta capa son HTML 5, CSS 3, Javascript y Bootstrap 4.

En la capa de lógica de negocios se lleva a cabo el procesamiento de la información, y se entrega gran parte del comportamiento del sistema según las solicitudes del usuario. Además, esta capa mantiene la conexión a la base de datos para recuperar y almacenar la información. Las tecnologías que utiliza esta capa son el framework Django[14], para desarrollo de aplicaciones web en Python, y Python para el funcionamiento de la lógica propia de la aplicación.

Según lo indicado en la memoria de Pablo Miranda[2], el motivo por el que se escogió Django, además de ser un framework escrito en Python y un lenguaje ampliamente utilizado durante la época del desarrollo de la plataforma legada, era que para crear una plataforma web en línea, se tenían Django y Flask[15] como opciones, y Django es un framework full-stack, es decir, posee herramientas para desarrollar las 3 capas descritas en el modelo de la Figura 2.1. Flask, a diferencia de Django, no provee herramientas que ayuden en la conexión con bases de datos[16], punto importante para el desarrollo de la plataforma legada. Por estas razones se decidió originalmente trabajar con Django por sobre otros frameworks, lo que no significó una dificultad para el memorista puesto que este maneja el framework.

Por último, en la capa de persistencia de datos, se guarda la información para que el sistema funcione apropiadamente, soportando desde el sistema de usuarios hasta la información de los clientes y los reclamos. Esta capa utiliza PostgreSQL como motor de bases de datos, que cuenta con un amplio uso y compatibilidad con el framework Django, y permite una gestión apropiada de los datos utilizados por este sistema.

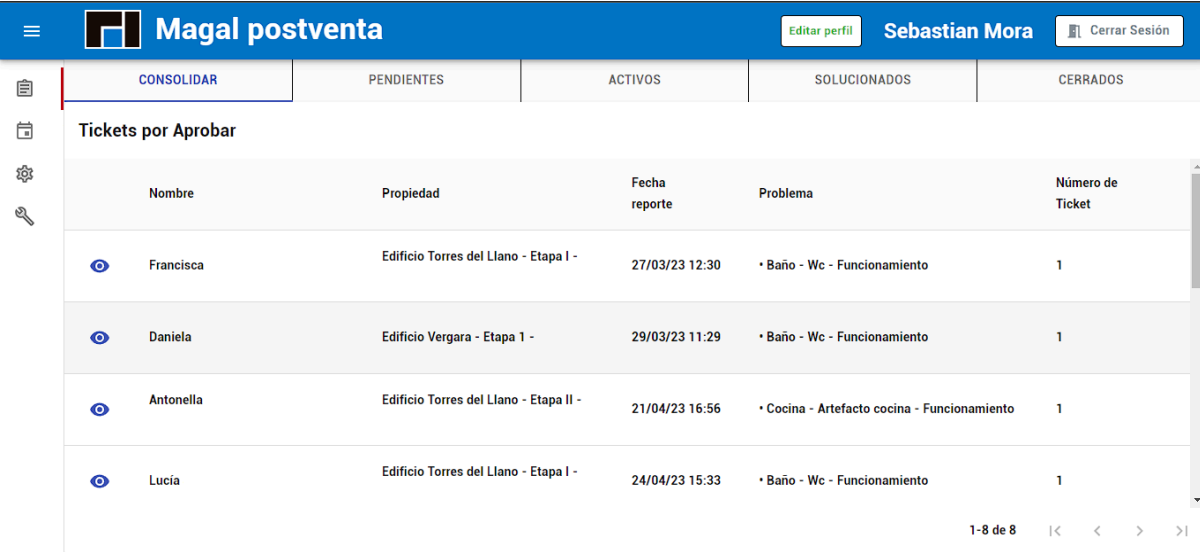
2.2. Descripción del Front-End

El front-end del sitio Web está desarrollado utilizando la librería React[3] de Javascript. La plataforma cuenta con una página principal, a la cual el usuario llega si no ha ingresado al sistema. Esta página le permite realizar dos acciones: 1) crear un ticket para reportar uno o más problemas, y 2) abrir una ventana para ingresar al sistema y tener acceso a las otras funcionalidades.

Actualmente, el sistema no le pide al usuario autenticarse para enviar un ticket, por lo que cualquier persona puede reportar uno o más problemas en una propiedad que no le corresponde. Esto representa una dificultad importante para Magal, ya que por política de la empresa se deben revisar todos los tickets reportados. Para cada ticket, los empleados de Magal deben corroborar que la información proporcionada por la persona sea verídica, y que ésta tenga relación con una propiedad que la empresa maneja, lo cual lleva tiempo y un esfuerzo no menor.

Por otro lado, no existe una opción para registrarse en el sistema, pues la opción proporcionada por la ventana de ingreso sólo permite autenticar usuarios, y recuperar la contraseña en caso de que se haya olvidado. Las cuentas nuevas se crean desde una pestaña a la cual sólo tienen acceso los usuarios administradores, y se crean para usuarios con roles dentro de Magal, como por ejemplo, ingenieros de un proyecto, y miembros de un equipo técnico, entre otros. Cuando un usuario (personal de Magal o de uno de sus edificios) ingresa al sistema, accede a la vista principal, la cual varía según su rol.

En la Figura 2.2 se muestra la vista que le muestra el sistema a un usuario administrador después de autenticarse. La página se divide en cuatro vistas, con diferentes pestañas. La primera corresponde a los tickets del sistema, los cuales se muestran como una lista, y se puede ver el detalle de cada uno de estos, y realizar diversas acciones sobre ellos.



The screenshot shows the 'Magal postventa' system interface. At the top, there is a navigation bar with a menu icon, the logo, the name 'Magal postventa', a profile button for 'Sebastian Mora', and a 'Cerrar Sesión' button. Below the navigation bar, there are five tabs: 'CONSOLIDAR' (selected), 'PENDIENTES', 'ACTIVOS', 'SOLUCIONADOS', and 'CERRADOS'. The main content area is titled 'Tickets por Aprobar' and contains a table with the following data:

Nombre	Propiedad	Fecha reporte	Problema	Número de Ticket
Francisca	Edificio Torres del Llano - Etapa I -	27/03/23 12:30	• Baño - Wc - Funcionamiento	1
Daniela	Edificio Vergara - Etapa 1 -	29/03/23 11:29	• Baño - Wc - Funcionamiento	1
Antonella	Edificio Torres del Llano - Etapa II -	21/04/23 16:56	• Cocina - Artefacto cocina - Funcionamiento	1
Lucía	Edificio Torres del Llano - Etapa I -	24/04/23 15:33	• Baño - Wc - Funcionamiento	1

At the bottom right of the table, there is a pagination indicator '1-8 de 8' and navigation arrows.

Figura 2.2: Vista principal de un usuario administrador al ingresar al sistema.

En esta interfaz, la pestaña “Consolidar” lista los tickets enviados por los usuarios, los cuales están a la espera de ser aprobados, para luego poder darles seguimiento a los mismos. Por su parte, la pestaña “Pendientes” lista los tickets ya aprobados, que están a la espera de que un empleado de Magal agende una cita utilizando un calendario digital de la misma página. La pestaña “Activos” corresponde a los tickets con una (o más) visitas agendadas, y que también están a la espera de ser realizadas. La pestaña “Solucionado” lista los tickets que tienen todos sus problemas resueltos, pero deben ser verificados por un usuario ingeniero, y por lo tanto

no pueden marcarse como cerrado. La pestaña “Cerrados” muestra los tickets ya resueltos y verificados por el personal de la empresa.

La segunda vista (Figura 2.3) permite a los usuarios ver los tickets pendientes en una lista. Al seleccionar uno, se le muestra al usuario un calendario digital con el cual se puede agendar una visita para revisar lo reportado en el ticket. Sin embargo, las visitas también se pueden agendar desde la primera vista, por ejemplo, al clicar el ticket para ver su detalle. En esta vista hay pestañas que muestran las visitas del día, y las visitas pendientes que no han sido finalizadas.

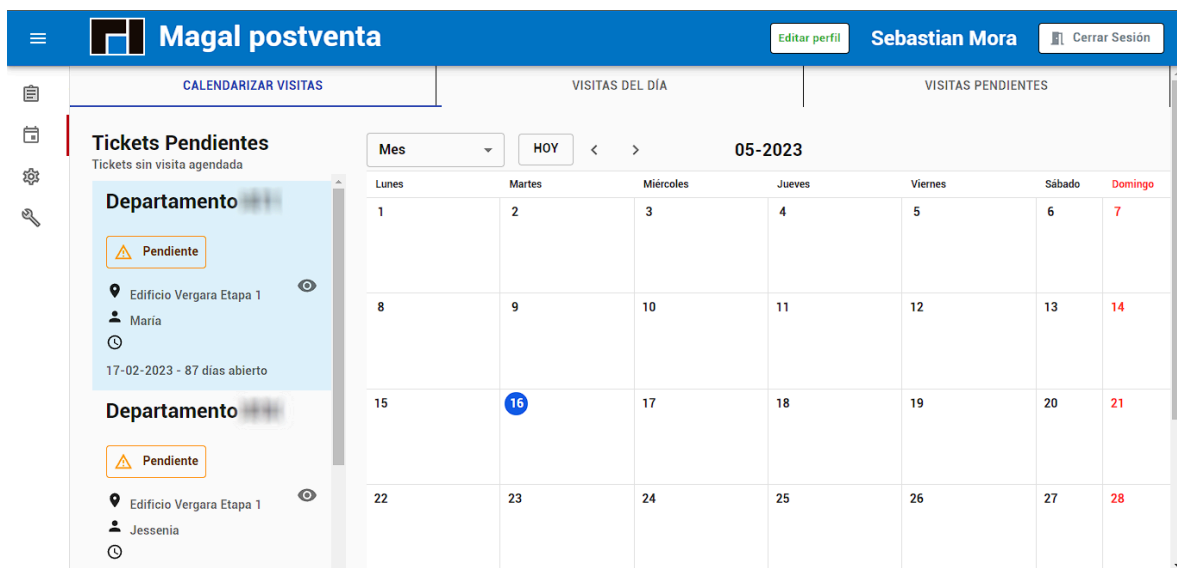


Figura 2.3: Vista de tickets pendientes y calendario para las visitas.

La tercera vista (Figura 2.4) corresponde a una configuración con acceso, disponible únicamente para usuarios administradores. Ésta cuenta con diversas pestañas para administrar usuarios, proyectos y equipos de trabajo.

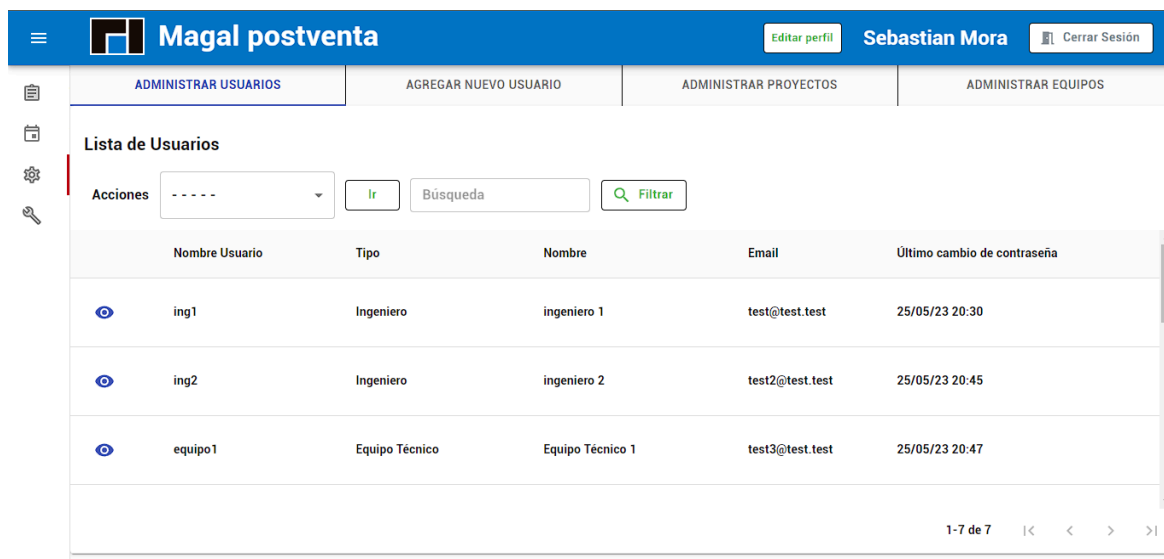


Figura 2.4: Vista de configuración, visible para usuario Administrador

La primera pestaña lista los usuarios existentes, que actualmente son sólo los empleados de Magal, los cuales se pueden seleccionar para editar su información o eliminarlos. La segunda pestaña permite a un administrador agregar un nuevo usuario, ingresando información básica y su correo para recibir la contraseña de ingreso. Actualmente, esta es la única forma de crear nuevos usuarios.

La tercera pestaña lista los proyectos existentes en la base de datos, es decir, los edificios. Dichos proyectos se pueden seleccionar y editar su información básica; por ejemplo, su nombre, comuna, dirección (ubicación) y el usuario ingeniero que está a cargo. Las propiedades asociadas al edificio no se pueden visualizar ni editar.

En dicha interfaz existe un botón para agregar un nuevo proyecto, pero esta acción sólo crea un proyecto con la información básica, y no se puede agregar ni asignar propiedades ni información de los clientes relacionados a éstas. Por lo tanto, actualmente no es utilizado por Magal, ya que no cumple la función esperada.

Por último, la cuarta pestaña (“*administrar equipos*”) lista los equipos existentes, que corresponden a agrupaciones de un ingeniero y más usuarios colaboradores. Los usuarios que pertenecen a un equipo tienen acceso a la información de los edificios que gestiona el ingeniero a cargo del mismo.

La última vista (Figura 2.5), lista las visitas pendientes, pero a diferencia de la segunda vista, en ésta se pueden iniciar las visitas y subir fotos que corresponden al proceso realizado. Esta vista la utilizan usuarios de tipo “equipo técnico”. En este caso, solo se les muestra las visitas agendadas en edificios a los cuales el usuario es parte de un grupo con el ingeniero a cargo. En el caso de un usuario “administrador”, éste tiene acceso a todas las visitas.

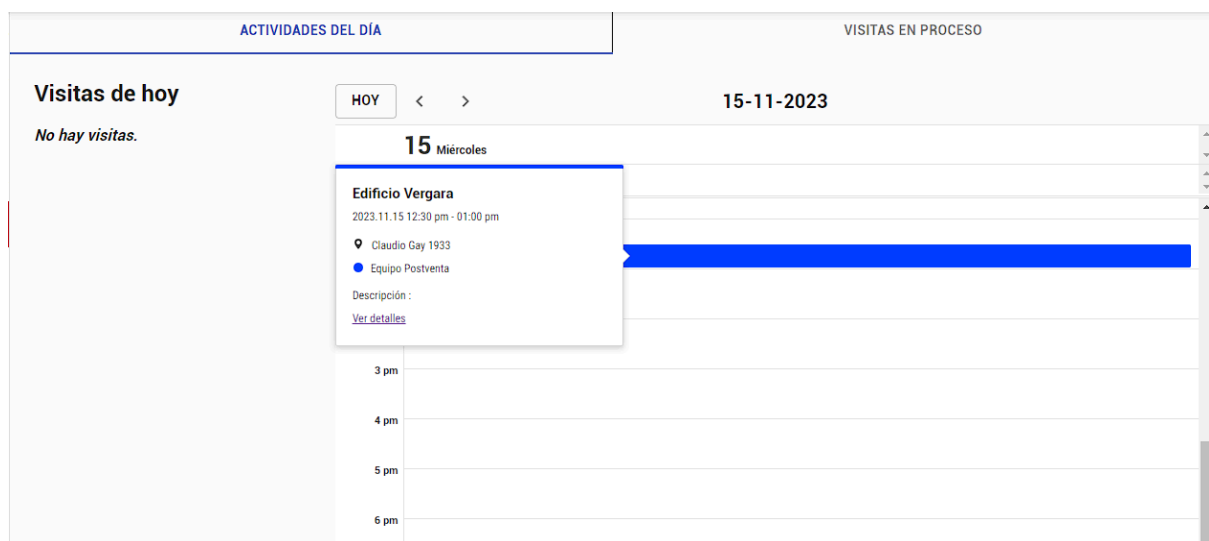


Figura 2.5: Vista de usuario equipo técnico.

Para diferenciar los distintos tipos de usuario soportados por el sistema, a continuación se describe cada uno de ellos y los permisos que tienen para operar sobre los datos disponibles.

- *Usuario Administrador*: Este usuario tiene acceso a todas las vistas, pestañas y puede realizar cambios sobre los tickets, visitas, usuarios y proyectos.
- *Usuario Encargado Post-venta*: Este usuario tiene acceso a la vista de tickets y visitas, puede aprobar tickets y agendar visitas a tickets de todos los edificios.
- *Usuario Ingeniero*: Este usuario puede ver los tickets asociados a los edificios que él tiene a cargo, y realizar acciones sobre dichos tickets.
- *Usuario Supervisor*: Puede ver los tickets, pero no realizar acciones sobre ellos.
- *Usuario Equipo Técnico*: Estos usuarios solo pueden ver la pestaña de visitas pendientes, iniciarlas y finalizarlas. Las visitas que se muestran corresponden a las de los edificios que el usuario está vinculado (por un grupo).

Además, se vio que existía una vista destinada a utilizar las estadísticas del sitio, pero como se explicó en la memoria de Pablo Miranda [2], ésta se descartó debido a limitaciones en su operación. Esta vista actualmente está oculta, y se pretende restaurarla una vez completada su funcionalidad; ese trabajo será realizado como parte de esta memoria.

2.3. Descripción del Back-End

El back-end de la plataforma está desarrollado utilizando el framework Django REST. Para comprender este componente, se realizó un análisis a los diferentes métodos y endpoints del mismo, y a sus conexiones con el front-end. Esto permitió entender cómo funciona la solución y cómo están enlazados sus componentes. A partir de esto, se describen a continuación elementos importantes que sirvieron para la resolución de los problemas planteados y resueltos en este trabajo de título.

Primeramente, los permisos de usuario existen tanto en back-end como en front-end. En el back-end, los diferentes métodos utilizados para recibir listas de tickets, visitas, usuarios, etc., tienen permisos asociados. La clase *User*, que deriva de *AbstractUser* y es utilizada para el ingreso de usuarios al sistema. Ésta cuenta con un campo *type*, el cual permite distinguir el tipo de usuario según un código. Los códigos se corresponden con los tipos de usuarios descritos en la sección 2.2, y son los que otorgan permisos a los distintos métodos.

Si un usuario realiza una solicitud a un método para el cual no tiene permisos asignados, esta será rechazada. Por ejemplo, si un usuario de tipo *ingeniero* intenta invocar un método al cual solo los usuarios de tipo *administrador* tienen acceso, la solicitud será rechazada. A su vez, en el front-end, si un usuario que no es *administrador* intenta ingresar a la vista de configuración para administradores, éste será redirigido a una vista principal a la que tenga acceso, o en algunos casos, la vista no cargará. Esto ocurre para otros tipos de usuario, como

por ejemplo los del “Equipo Técnico”, que no pueden ingresar a otras vistas cambiando la URL, ya que son redirigidos a la vista principal a la cual tienen acceso.

Existe un método para agregar nuevos usuarios al sistema. Éste solo lo puede utilizar un usuario *administrador*, y es parte de una clase llamada *UserViewSet*, la cual es de tipo *ModelViewSet*. Esta clase cuenta con distintos métodos de filtrado y creación, entre otros. La clase sirve como guía para agregar una similar, para que un usuario que no sea personal de Magal (típicamente, un cliente) pueda registrarse en el sistema.

En relación a la creación de proyectos, existe una clase llamada *ProjectViewSet*, que contiene métodos para modificar o eliminar un proyecto, y principalmente, para agregar uno nuevo. Este último es de interés para resolver el problema que presenta Magal, al no poder agregar proyectos con toda la información correspondiente, para lo cual se deberá modificar y adaptar el método para lograr dicho objetivo.

Además, hay diferentes vistas y clases que entregan información de acuerdo al tipo de usuario que lo solicita. Por ejemplo, hay clases que utilizan *ListAPIView* para entregar un ticket o un listado de estos. Existe una clase llamada *GetTicketByIDandRut* que recibe 2 parámetros para buscar un ticket que coincida con ellos, y luego retornarlo. Esta clase es utilizada en una vista que no requiere permisos para acceder, y es aquella a la cual los clientes acceden para ver el estado de un ticket que reportaron. Una clase con una función similar se puede utilizar en una vista para usuarios tipo “cliente”, que ya están registrados e inician sesión para ver sus tickets reportados.

Por último, se revisaron las URLs, serializers y el modelo de datos existente, para tener una mejor idea de cómo están conectados a las clases y vistas. Finalmente, se empezó a diseñar la solución que permite resolver los problemas, y dividirla en componentes más simples que permiten una resolución fluida de los tickets.

2.4. Descripción del Modelo de Datos

El modelo de datos de la plataforma legada fue diseñado originalmente pensando en la eficiencia de las operaciones más recurrentes de post venta. El tipo de datos que trabaja el sistema son en su mayoría nativos de Django, como datos textuales, números, fechas y archivos. Sin embargo, hay dos tipos de archivos para los cuales se necesitaban librerías adicionales. Primeramente, para el manejo de imágenes, como la firma de los clientes y las fotografías adjuntas en los reclamos y visitas, se necesitó la librería *Pillow* [4] de Python.

Para el manejo de archivos Excel, formato de preferencia de Magal, se necesitó utilizar la librería *Pandas* [5] de Python. Esto fue utilizado para realizar la carga masiva de datos (etapa inicial) , que la llevó a cabo una persona externa a Magal.

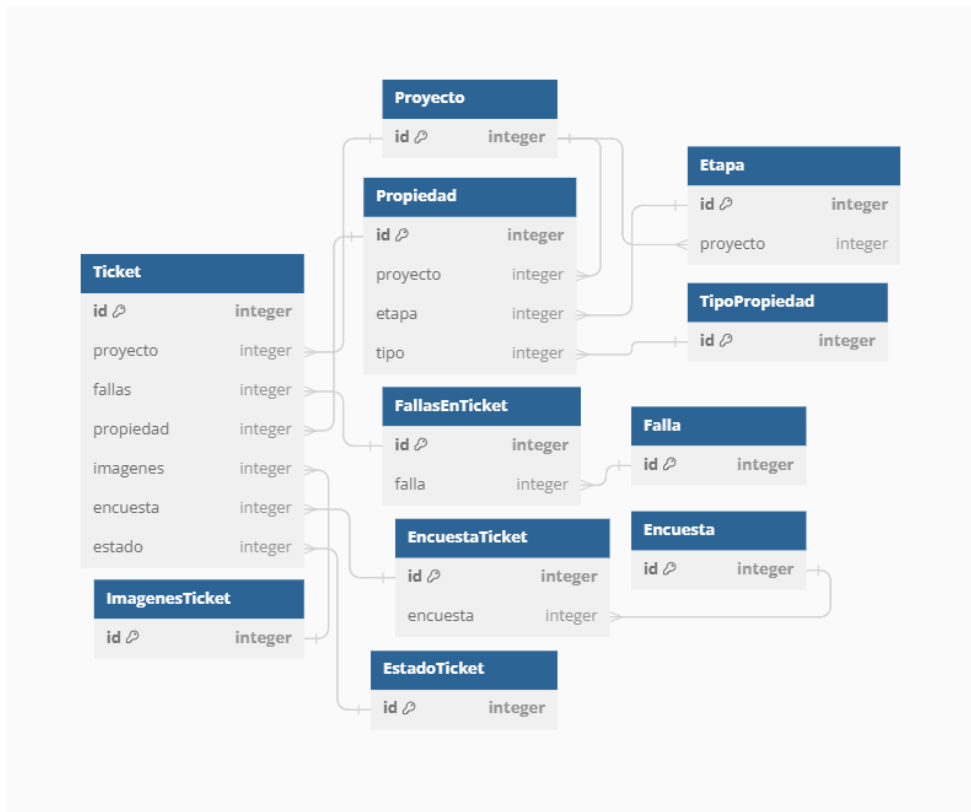


Figura 2.6: Tablas principales del modelo de datos de la plataforma legada

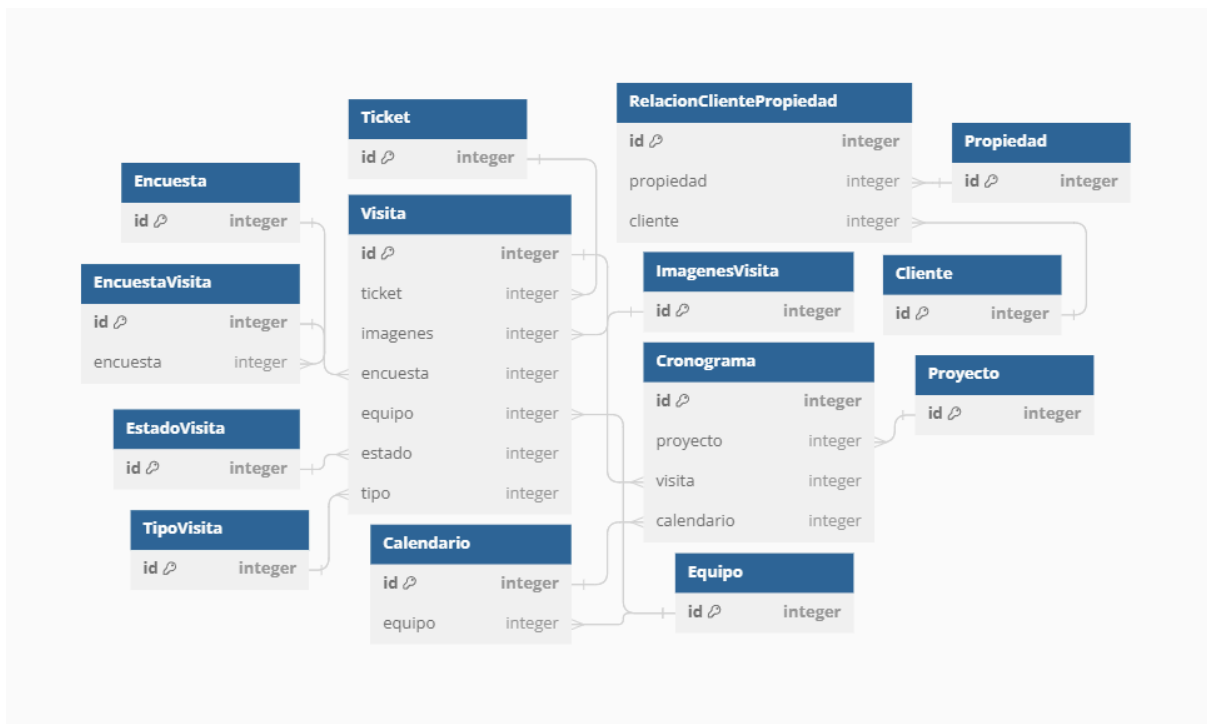


Figura 2.7: Tablas principales restantes del modelo de datos de la plataforma legada

Las tablas mostradas en las Figuras 2.6 y 2.7 corresponden a una versión simplificada del modelo de datos de la plataforma legada, donde se omiten las tablas relacionadas al control

de usuarios, y solo se muestran los atributos con llaves foráneas a otras tablas. A continuación se describe brevemente cada tabla.

Cliente: En esta tabla se guarda la información de los clientes de la empresa, es decir, usuarios que poseen una propiedad de la inmobiliaria. Se almacenan datos personales como el nombre completo, RUT, fecha de nacimiento, y datos de contacto.

Proyecto: En esta tabla se aloja la información básica de los edificios que ha construido Magal. Se guarda el nombre del proyecto, la comuna, el ingeniero a cargo.

Etapas: Esta tabla tiene una llave foránea a *Proyecto*, y se almacenan el nombre, la dirección, número de etapa, y una fecha de recepción municipal. Hay casos en donde un edificio (proyecto) tiene más de una torre, por lo que era necesario para Magal tener esta separación.

Propiedad: Esta tabla posee una llave foránea tanto a *Proyecto* como a *Etapas*, para saber a cuál edificio y torre pertenece. Se guardan además datos como el código (el número de un departamento, por ejemplo), el precio, y el tipo de propiedad, entre otros.

Tipo de Propiedad: Esta tabla almacena un código y etiqueta, los cuales permiten distinguir el tipo de propiedad, para saber si es un Departamento, Estacionamiento, Bodega, Local o Áreas Comunes.

Relación Cliente Propiedad: Esta tabla posee una llave foránea a cliente y propiedad, los cuales vincula a través de una relación descrita por un campo de texto, por ejemplo, "Dueño".

Falla: Esta tabla cuenta con llaves foráneas a las tablas *ZonaFalla*, *TipoFalla* y *ObjetivoFalla*, de forma tal que se almacena la combinación de zona macro del problema, su tipo y en específico la ubicación de éste.

Falla en Ticket: Cuenta con una llave foránea a *Falla*, y se almacena un booleano para saber si fue solucionado y un contador de iteraciones, es decir, el número de veces que se repite el problema para la misma propiedad.

Ticket: Esta tabla almacena toda la información de un reclamo. Cuenta con campos para los problemas, su estado, imágenes, comentarios, datos del cliente que realiza el reclamo, además de campos adicionales que son útiles para las acciones que realiza el personal de Magal.

Visita: Guarda toda la información de una visita. Allí se almacenan datos como el ticket correspondiente, su estado, las fechas, cronograma, las imágenes, información de quien la realiza y sus comentarios.

Encuesta: Esta tabla almacena datos de una encuesta, cuenta con campos como el tipo para identificar si es de visita o de ticket, una lista de preguntas e instrucciones.

Cronograma: En esta tabla se almacenan campos relevantes de una visita programada, como la fecha de inicio y término, el edificio y su ubicación, y una llave foránea a *Calendario*.

Calendario: Ésta cuenta con una llave foránea a *Equipo* y campos para almacenar datos de colores. Estos colores son utilizados al momento de elegir el equipo a cargo de una visita, a modo de diferenciarlos.

Equipo: Esta tabla cuenta con una llave foránea a *Usuario*, y un campo para el nombre del equipo. Almacena un usuario a cargo, y usuarios que son miembros del mismo. Esto se utiliza para otorgar los permisos para realizar visitas.

Originalmente el modelo de datos se había pensado con tablas autocontenidas, para así reducir el acceso a otras tablas. Sin embargo, el modelo estudiado, correspondiente a la situación legada, contiene tablas con diversas llaves foráneas para mantener una menor complejidad en el manejo de la información.

2.5. Estado del Arte

Actualmente en el mercado existen muchos productos de tipo CRM. Existen tanto sistemas genéricos con funcionalidades básicas, como sistemas con servicios especializados en otros rubros. Se investigaron diversos CRMs para analizar si era más conveniente para Magal contratar un nuevo servicio, o realizar una extensión sobre el sistema actual, que terminó siendo el objetivo de este trabajo de memoria. A continuación se describen tres de los CRMs investigados, sobre los cuales se señala precio y características.

PipeDrive CRM. La empresa PipeDrive [6] está especializada en ventas, y el sistema ayuda a aumentar las redes de contacto a través de publicidad, con el objetivo de cerrar negocios con nuevos usuarios. Esta empresa ofrece su servicio más básico por un precio aproximado de 13 dólares al mes por usuario. Considerando la cantidad de usuarios trabajando en Magal, se debería pagar por lo menos 100 dólares al mes por el plan más básico, que trae consigo varias limitaciones. El precio para acceder a un plan más avanzado eleva el costo hasta más de 500 dólares por mes, pero limita a Magal a agregar los servicios especializados que esta empresa necesita.

Monday sales CRM. La empresa Monday [7] cuenta con una variedad de servicios, incluyendo desarrollo de software, marketing, gestión de proyectos, y postventa, entre otros. Su servicio de CRM cuenta con distintos planes, donde el más básico tiene un costo de 30 dólares al mes por usuario, mientras que el plan profesional supera los 70 dólares por usuario. Sin embargo, los servicios de post-venta solo están disponibles en el plan

corporativo, que cuenta con más características que el profesional, y por ende, resulta mucho más costoso.

BReal CRM. La plataforma de Administración de Mantenimiento de BReal [8] es un software que permite la recepción de tickets de requerimientos, pero su enfoque es la administración de proyectos. Este software cuenta con una mayor variedad de características que se alejan de la postventa. El coste anual de este CRM por tres usuarios es de 1.620 dólares, por lo que mensualmente Magal pagaría una cifra mayor a 400 dólares considerando la cantidad de usuarios.

Sumado a las opciones anteriores, se hizo una investigación de otros CRM existentes en el mercado. En todos los casos, las licencias que se deben pagar por mes significan un costo elevado, y muchas de las funciones que ofrecen estos sistemas no serían aprovechadas por Magal, tomando en cuenta que la empresa solo requiere de un sistema de postventa útil. Por esta razón, se considera que es conveniente para Magal continuar utilizando su CRM actual, y extenderlo para agregar las nuevas funcionalidades que necesitan, sin incurrir en gastos mensuales por licencias si se decidiera optar por un nuevo software CRM.

El CRM desarrollado para Magal tiene las características de un CRM operacional. Éste está enfocado en optimizar los procesos relacionados con el cliente, unificando las actividades de distintos tipos de usuarios (con diferentes permisos) a través de interfaces comunes, que permiten una mayor colaboración y productividad de los usuarios.

En el trabajo de memoria de Pablo Miranda [2] se explica que los CRMs investigados no eran lo suficientemente adecuados para Magal como empresa, razón por la cual el CRM actual fue diseñado y creado a la medida. Se hizo uso de la documentación del mismo, para integrar los cambios y modificaciones a dicho sistema, lo cual no fue una tarea trivial.

2.6. Investigación de Librerías

A fin de tener alternativas para el desarrollo del módulo de estadísticas, se investigaron varias librerías de React que podían ser utilizadas en la implementación de esta extensión. Al momento de investigar, se destaca D3, la cual corresponde a una librería popular de JavaScript para visualizar datos, siendo ésta una de las opciones investigadas.

Recharts. Recharts es una librería para gráficos construida con D3 y React, y cuenta con licencia MIT. Ésta es muy popular, cuenta con aproximadamente 19.000 estrellas en Github [9], y documentación sencilla de entender. Este aspecto facilita el aprendizaje del uso de la misma, y acelera el proceso para entenderla y agregarla al proyecto. Lo útil de esta librería es que se puede incluir simplemente con un ‘import’, y reutilizar los componentes para crear gráficos, siguiendo el mismo principio de React.

Victory. Victory es otra librería construida con D3 y React, que cuenta con una variedad de gráficos que se pueden personalizar. Esta librería cuenta con código de libre uso y

una documentación robusta, lo que la hace amigable con los principiantes y fácil de empezar a usar, por lo que es una buena opción a considerar.

React-charts. Esta es una librería altamente responsiva, cuenta con un modelo de datos flexible, también está construida con D3, tiene licencia *MIT* y soporta una variedad amplia de gráficos.

Las anteriores son las opciones más destacables que se encontraron. Dadas las recomendaciones de diferentes sitios Web, y considerando la descripción de cada una, se decidió incluir *Recharts* en el proyecto como librería para visualizar y generar gráficos. De esta forma, la próxima persona que se haga cargo de modificar y agregar más estadísticas al proyecto, puede utilizar esta librería para agregar nuevos tipos de visualizaciones. Gracias a su documentación detallada, esta labor no será un trabajo difícil.

Capítulo 3

Concepción de la Solución

En los capítulos anteriores se habló sobre la situación de Magal y las dificultades que presentaba el uso del sistema legado. Para completar los objetivos planteados, los cuales resuelven dichos problemas, en este capítulo se detallan los requisitos y la solución diseñada como parte de esta memoria.

3.1. Principales Requisitos de la Solución

Se describen a continuación los principales requisitos que motivaron el desarrollo de este trabajo de memoria.

- *Carga Masiva:* Este requisito apunta a que se pueda consumir un archivo excel, con un formato predeterminado, con el fin de realizar un proceso *ETL* (Extraer, transformar y cargar) y agregar el contenido a la base de datos.
- *Registro de Usuarios:* Se requiere que una persona pueda llenar un formulario con sus datos, a fin de que sean validados por el sistema y crear una cuenta para ingresar al mismo.
- *Manipulación de Datos:* Este requisito se refiere a la capacidad de Magal de manejar los datos existentes, a fin de editar y eliminar aquellos relevantes, como información de proyectos, propiedades y relación de clientes con éstas.
- *Panel de Clientes:* Este requisito se refiere a la adición de una vista para los usuarios clientes, donde puedan realizar diversas acciones, como crear nuevos tickets, ver sus tickets y encuestas existentes, y editar sus datos de perfil.
- *Aprobación de Clientes:* Magal requiere una vista que liste los clientes registrados que no fueron validados automáticamente por el sistema, con el fin de verificar sus datos y validarlos de forma manual.
- *Panel de Estadísticas:* Se requiere contar con una vista que entregue información general sobre los tickets, visitas, problemas y otros datos, que permitan a Magal mejorar la toma de decisiones estratégicas.

3.2. Diseño de la Planilla Excel

El primer objetivo específico definido se divide en dos partes: 1) el diseño de una planilla en formato Excel, para realizar la carga de nuevos proyectos por parte de Magal, y 2) el desarrollo de un servicio que consuma los datos de la planilla y actualice la base de datos del sistema. Para completar estos pasos se realizaron reuniones con Magal, fundamentalmente para acordar el formato de dicha planilla, las columnas de cada hoja, y lo que representa cada una.

Se acordó tener una planilla para la carga de datos que se compone de 4 hojas, de las cuales se recibirá la información para agregar los nuevos proyectos, sus torres, propiedades, clientes, y la información relacionada a los clientes y sus relaciones con las distintas propiedades. A continuación se muestran las columnas de cada hoja diseñada para la planilla de Excel, que tras la aprobación de Magal, fueron las utilizadas para agregar nuevos edificios.

La primera hoja posee la información de los edificios. Las columnas de esta hoja contienen la información base de los nuevos proyectos y etapas, y se pueden visualizar en la Tabla 3.1 a continuación, donde se explica brevemente cada una.

<i>Columna</i>	<i>Nombre</i>	<i>Descripción</i>
A	Nombre	Nombre del edificio
B	Comuna	Comuna donde se ubica el edificio
C	Nombre Etapa	Nombre de la torre
D	Fecha Recepción Etapa	Fecha para plazo de post venta sobre la etapa
E	Dirección Etapa	Dirección donde se ubica la etapa
F	Número Etapa	Enumeración de la etapa
G	Proyecto+Etapa	Identificador único del proyecto y etapa

Tabla 3.1: Columnas de la hoja 1 de la planilla Excel

La última columna combina los valores de las columnas A, B, C y F. Ésta tiene como fin identificar las propiedades que pertenecen a un edificio determinado. Con la información de esta hoja, por cada fila se crea una nueva entrada en la tabla *Project*, siempre que no exista uno con el mismo nombre, usando los datos de las primeras 2 columnas. Luego se crea una nueva entrada en la tabla *Stage* (Etapa) con la información de las columnas restantes.

La segunda hoja contiene la información de las propiedades de cada edificio (Tabla 3.2). La columna A coincide con la columna G de la primera hoja, de esta forma se puede obtener el proyecto y etapa a las cuales pertenece la propiedad. La columna C indica si la propiedad es un departamento, estacionamiento, bodega o local.

<i>Columna</i>	<i>Nombre</i>	<i>Descripción</i>
A	Proyecto+Etapa	Identificador del proyecto y etapa
B	Código	Comuna donde se ubica el edificio
C	Tipo	Tipo de propiedad
D	Piso	Piso donde se ubica la propiedad
E	Orientación	Tipo de orientación de la propiedad
F	M2 Producto	Metros cuadrados de la propiedad
G	M2 terreno	Cantidad de metros cuadrados de terreno
H	Identificador Propiedad	Identificador único de la propiedad

Tabla 3.2: Columnas de la hoja 2 de la planilla Excel

Con la información de estas columnas, por cada fila se crea una nueva entrada en la tabla *property*, y utilizando el identificador de la columna A, se asocia el proyecto y etapa correspondientes. La columna H le otorga un identificador a la propiedad, combinando las columnas A y B.

La tercera hoja (Tabla 3.3) contiene los datos de los clientes. Las columnas contienen la información de cada cliente que será agregado, y por cada fila se crea una entrada en la tabla *client*, con los datos correspondientes. La columna H copia el valor de la columna D, y se utiliza para identificar los clientes al momento de crear las relaciones entre cliente y propiedad.

<i>Columna</i>	<i>Nombre</i>	<i>Descripción</i>
A	Nombre	Nombres del cliente
B	A. Paterno	Apellido paterno del cliente
C	A. Materno	Apellido materno del cliente
D	Rut	Rol único tributario del cliente
E	Género	Género del cliente
F	Teléfono	Teléfono de contacto del cliente
G	Correo	Correo de contacto del cliente
H	Identificador Cliente	Campo equivalente al Rut

Tabla 3.3: Columnas de la hoja 3 de la planilla Excel

La cuarta y última hoja (Tabla 3.4) contiene la relación de los clientes con las propiedades. La primera columna es equivalente a la última columna de la hoja 3, es decir, el *rut* del cliente. La segunda columna es la fecha de entrega de la propiedad al cliente, asignada por Magal, y la tercera columna es equivalente a la última columna de la hoja 2.

<i>Columna</i>	<i>Nombre</i>	<i>Descripción</i>
A	Identificador Cliente	Identificador del cliente
B	Fecha Entrega	Fecha en que se entrega la propiedad al cliente
C	Identificador Propiedad	Identificador de la propiedad

Tabla 3.4: Columnas de la hoja 4 de la planilla Excel

A partir de los datos de esta hoja se agregarán las relaciones *cliente-propiedad* al sistema, las cuales permitirán a los clientes que se registren en el sistema poder reportar problemas en sus propiedades vinculadas a partir de esta relación.

Así, con los datos obtenidos de las 4 hojas de esta planilla, se agregan los nuevos edificios, creando así nuevos proyectos, etapas, propiedades, clientes y las relaciones de éstos con las propiedades.

3.3. Diseño del Módulo de Carga de Datos

La segunda parte del primer objetivo corresponde a crear una vista que permita la carga masiva de datos, haciendo uso de la planilla descrita en la sección anterior. Dado que en el sistema legado existe un recuadro para agregar nuevos proyectos, ubicado dentro de la pestaña para administrar los proyectos existentes, este se eliminará puesto que solo permite agregar proyectos sin información de torres, propiedades ni relaciones.

Se creará una nueva pestaña en la vista de administración, y dentro de esta, se agregará un recuadro que permita al usuario cargar un archivo con extensión *.xlsx*, a fin de utilizar un archivo con datos sobre la planilla diseñada anteriormente. Se muestra a continuación un bosquejo de esta pestaña.

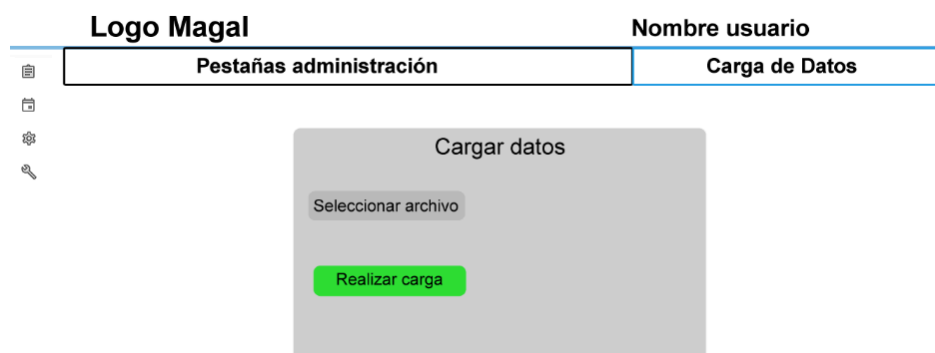


Figura 3.1: Bosquejo de la interfaz para la carga de datos.

En la Figura 3.1 se muestra que esta pestaña estará junto a las otras existentes en la vista de administración. El recuadro principal de esta pestaña permitirá al usuario seleccionar un archivo para ser subido, y con un botón se da inicio a la carga de datos utilizando dicho archivo. En este mismo recuadro se mostrarán los resultados obtenidos de la carga.

De forma adicional, según lo conversado con Magal, se tendrán 2 recuadros adicionales que siguen el mismo formato de carga, pero cumplen funciones diferentes. Uno es solo para cargar clientes, mientras que el otro es solo para cargar relaciones cliente-propiedad. Esto es dado el flujo con el que Magal agrega la información de un nuevo edificio a la base de datos, el cual debe ser actualizado cada cierto tiempo con nuevos clientes y relaciones.

Además, en esta vista se tendrá la opción de descargar la planilla, para así tener acceso rápido a ella. La extracción y procesamiento de los datos del archivo subido se explicará en una sección aparte en este mismo capítulo.

3.4. Diseño del Módulo de Estadísticas

El segundo objetivo específico de la memoria corresponde al diseño e implementación de un módulo de estadísticas, que entregue información sobre puntos relevantes a Magal para la toma de decisiones estratégicas. Para esto, se agregará una nueva vista con su url respectiva en el front-end. Dentro de esta vista, se agregarán opciones para que el usuario pueda visualizar diferentes estadísticas, utilizando una librería de React mencionada en el capítulo anterior. De la misma forma, se debe agregar un método en el back-end en el cual se realicen cálculos y ordenar los resultados para que sean enviados y visualizados en la interfaz correspondiente.

3.4.1. Diseño de la Interfaz

A continuación se muestra un bosquejo de la vista de estadísticas (Fig. 3.2), a la cual se podrá acceder mediante el icono característico en la barra lateral izquierda, o en su defecto ingresando a través de una url.

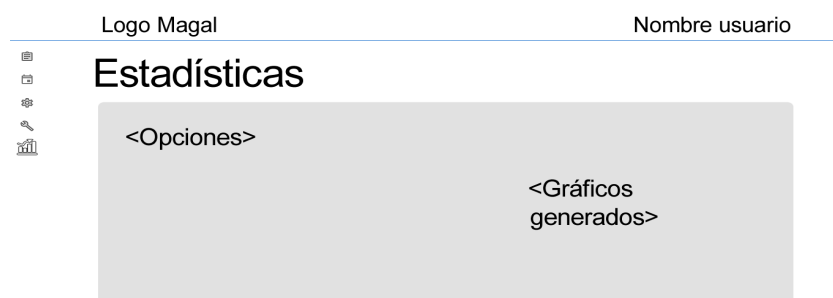


Figura 3.2: Bosquejo de interfaz para la vista de Estadísticas

En esta vista se permitirá a Magal ver los datos relevantes que seleccionen. Además, dado que se utilizará la librería *Recharts* de React (que posee buena documentación), cualquier persona que se encargue del proyecto a futuro no tendrá problemas para configurarla y agregar más opciones de gráficos o modificaciones. Se implementará la vista para que sea compatible con un navegador sobre un computador de escritorio, y se evaluará la posibilidad de que sea responsiva para celular y tablet.

Para lograr esto, en front-end se creó y agregó una nueva página, a la cual se puede acceder a través del menú lateral después de que un usuario de Magal inicia sesión (como se ve en la

Figura 3.2). Se asignaron permisos para que solo usuarios de tipo administrador puedan acceder a ella, se agregó un nuevo icono en la barra lateral, y se realizó la conexión con los métodos de back-end para recibir los datos. A partir de esto, se genera un gráfico por defecto en la vista para visualizar los datos obtenidos.

3.4.2. Cálculo de los Datos

En back-end se creó un método para calcular y ordenar los datos solicitados, según la estadística que el usuario quiera ver. A continuación se indica brevemente cómo se recuperan los datos, y el formato en que son entregados, según lo conversado con Magal.

Para el primer indicador, que es el tiempo promedio que demora Magal en resolver un problema, se traen todos los problemas de la tabla “Fault” en una lista, y se aplica un filtro en caso de que se quiera ver una zona y tipo en específico. Luego, se buscan todos los tickets cerrados donde los problemas asociados coinciden con alguno de la lista de problemas predefinidos. Magal decidió que el filtro debe ser sobre tickets en estado “cerrado”, ya que es posible que un ticket solucionado pueda volverse a abrir y así extender el tiempo en que un problema es resuelto. Finalmente, para cada ocurrencia del problema, se calcula la diferencia de tiempo desde que se reportó en el ticket, hasta que éste cuando fue resuelto. Además, se calcula el promedio y se ordenan los resultados. A partir de lo calculado se entregan los 15 problemas con mayor tiempo de resolución.

El segundo indicador, que es el tiempo que demora Magal en agendar una visita desde que se recibe un ticket, será dividido por proyectos, es decir, para cada edificio se entregará el tiempo promedio en agendar visitas, para saber cuáles son los que presentan más dificultades. Por cada elemento de la tabla “Project”, se filtran los tickets reportados en el proyecto actual; de estos tickets se excluyen aquellos rechazados. Para cada ticket, se obtienen las visitas asociadas a éste, ordenadas por fecha de creación. Luego, se calcula el tiempo transcurrido desde la fecha en que se reportó el ticket, hasta la creación de la primera visita. Finalmente, se calcula el promedio y se repite el proceso para el resto de los proyectos.

Para el tercer indicador, que es la cantidad de veces que se reporta un mismo problema, se recuperan (de la tabla “FaultInTicket”) todos los problemas asociados a ticket, y se cuentan las ocurrencias para cada uno. Luego se ordenan los resultados y se entregan los 15 problemas más reportados. En este caso también se considera un filtro para los problemas, si es que se quiere visualizar una zona o tipo específico de problema.

Por último, para determinar los tipos de departamentos con más problemas, Magal hizo notar que estos no están presentes actualmente en la base de datos, y en su reemplazo se trabajará con la orientación de los departamentos hasta que ese atributo sea agregado a futuro. Entonces para este último indicador, se recuperan todos los tickets de la tabla “Ticket”, y por cada uno se obtiene la orientación de la propiedad asociada. Si el ticket no cuenta con una propiedad, significa que es un reclamo sobre las áreas comunes del edificio, lo cual se agrega de todas formas a los resultados.

Cabe recalcar que todos los datos anteriores son calculados usando un rango de fechas, es decir, un período el cual será determinado por las entradas del usuario en la interfaz.

3.5. Diseño del Módulo de Registro de Usuarios

El tercer objetivo de esta memoria corresponde al rediseño de la página de reporte de tickets (reclamos), ubicada en la página de *landing*. Esta página solicita el inicio de sesión a los clientes antes de que estos puedan hacer uso del sistema. En el sistema legado no se pide un inicio de sesión a los usuarios, y cualquier persona puede reportar un ticket, a pesar de no ser residente de una vivienda.

Se modificará la vista principal, correspondiente al *landing* de usuario no logueado, para que el botón de “Reportar un problema” abra un recuadro que solicite al usuario iniciar sesión antes de realizar un reclamo. El siguiente es un bosquejo de cómo debe quedar esta vista con los cambios esperados (Fig. 3.3 y 3.4).

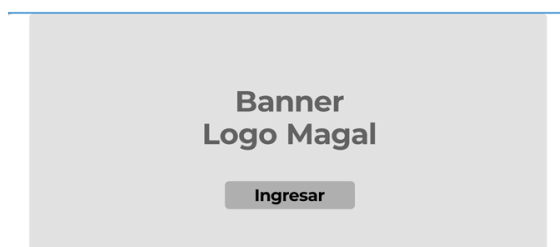


Figura 3.3: Bosquejo de interfaz para la vista principal

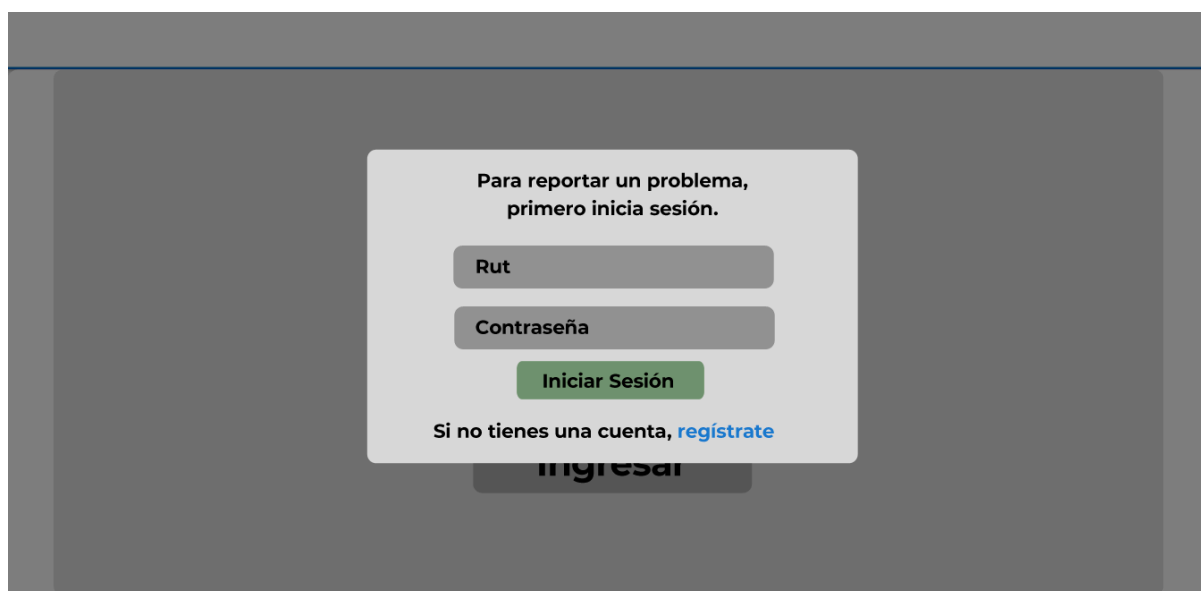


Figura 3.4: Bosquejo de interfaz para inicio de sesión en la vista principal

Los usuarios deberán iniciar sesión para reportar un nuevo problema en sus viviendas, así como también, para ver los problemas que ya han reportado. Además, se agrega la opción de registrarse a través de un formulario, si el usuario no posee una cuenta.

Cuando el usuario inicia sesión, será recibido por una nueva vista donde puede ver sus tickets, y un botón para reportar un nuevo problema. La figura 3.5 a continuación muestra un bosquejo de la vista que tiene un usuario cliente al momento de iniciar sesión.

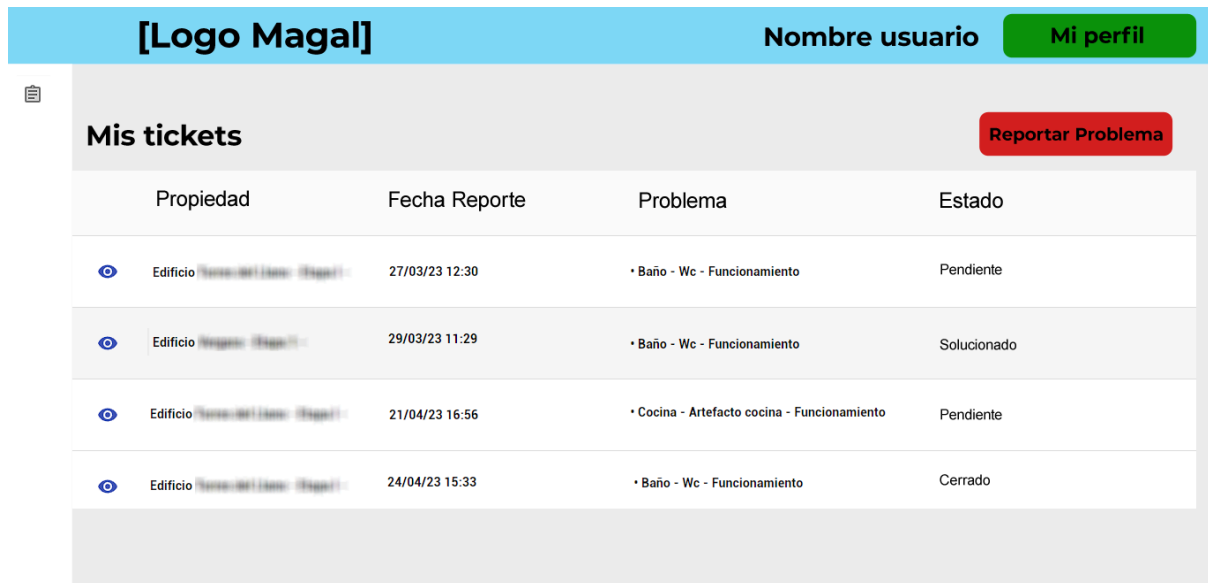


Figura 3.5: Bosquejo para vista de usuario cliente tras iniciar sesión

Además de los tickets creados, el usuario puede acceder al detalle de estos, y a las visitas que hayan sido agendadas sobre cada reclamo individual. La estructura de esta vista sigue la estética actual de la plataforma. Clickeando el botón “Mi Perfil”, el usuario puede cambiar la información básica de su cuenta, como su correo electrónico, su teléfono, entre otros. Al clickear el botón “Reportar Problema” se abrirá una ventana, la cual mostrará una adaptación del formulario de reclamo del sistema legado. Para este formulario, se quitará el paso que solicita los datos personales y de contacto del usuario, puesto que éste ya estará registrado en el sistema.

Finalmente, los cambios necesarios para completar este objetivo se dividen en tareas más concretas. A continuación se detallan los cambios a realizar en front-end:

- Modificar la vista del *landing* para que el recuadro emergente solicite inicio de sesión al usuario, y le ofrezca la opción de registrarse en el sistema.
- Agregar un nuevo recuadro con un formulario para registrarse como cliente en el sistema.
- Agregar una página para usuario cliente que inicia sesión, la cual le permite ver sus tickets, crear nuevos reclamos y editar su perfil.
- Conectar la vista anterior con el back-end para recibir todos los tickets reportados por el usuario.

- Agregar un recuadro emergente que despliegue los detalles básicos de un ticket de la lista, el cual se muestra al seleccionar el ticket correspondiente en la vista.

Luego, los cambios a realizar en back-end son los siguientes:

- Agregar un nuevo tipo de usuario “Cliente”.
- Agregar un método que crea un nuevo usuario tipo “Cliente” utilizando los datos enviados a través del formulario de registro.
- Agregar un serializer que entregue la información básica de un ticket, para que lo pueda visualizar un cliente.
- Agregar un método que entregue los tickets asociados a un usuario dado un identificador, utilizando el serializer descrito en la tarea anterior.

3.6. Proceso de Carga Masiva de Datos

Para el logro del primer objetivo específico se requiere, además del diseño de la planilla Excel explicado en el punto 3.2, el procesamiento del archivo subido utilizando dicha planilla en back-end. Esto se hace después de que un usuario lo envía a través de la interfaz en el sitio web.

Este archivo se carga en el sistema, y se hace uso de unas clases diseñadas previamente por otra persona que trabajó en el sistema, con el objetivo de realizar una carga masiva inicial. Estas clases se adaptaron para que pudieran procesar las hojas de la planilla, y que entreguen resultados que se puedan visualizar en la interfaz correspondiente.

Primeramente, como se ve en la Figura 3.6, en la línea 124 se inicializa la clase que procesa la hoja de proyectos y etapas. Luego, en la línea 125 se inicializa la clase que procesa la hoja de propiedades, y hace uso de la clase anterior. Luego, se inicializa la clase que procesa la hoja de clientes en la línea 126, y la clase que procesa la hoja de relaciones entre cliente y propiedad (en la línea 127). Esta última hace uso de las dos clases anteriores.

```
124 projects_processor = ProjectsSheetProcessor()  
125 properties_processor = PropertiesSheetProcessor(projects_processor)  
126 clients_processor = ClientsSheetProcessor()  
127 rcp_processor = RCPSheetProcessor(properties_processor, clients_processor)
```

Figura 3.6: Inicialización de las clases que procesan las hojas del archivo Excel.

Luego, se lee el archivo subido utilizando un método de la librería *Pandas*, y se verifica el nombre de la hoja para procesarla en la clase que corresponda. Para la hoja de proyectos, por cada fila se extrae la data de las columnas, y se verifica que el identificador no sea vacío; es decir, la columna “Proyecto + Etapa”. Seguidamente, se limpia la información y se guarda en un diccionario, y luego se intenta crear una nueva entrada en las tablas *Project* y *Stage* de la base de datos, utilizando el método *get_or_create* de Django.

De esta forma se verifica si existe un proyecto o etapa que coincida con los nombres correspondientes, y se indica a través del resultado de este método. La información del proyecto o etapa existente no es reemplazada, por lo que este proceso solamente agrega nuevas entradas, siempre que sea posible.

Si ya existe una etapa con su respectivo proyecto, se almacena el número de fila en un diccionario. En cambio si una fila presenta problemas, ya sea con los datos o porque el identificador está vacío, no se crea una entrada y se almacenan el número de la fila y el error correspondiente, y se continúa procesando las filas restantes. Así, una vez terminado el proceso, se le indica al usuario aquellas filas que presentaron errores, y aquellas que ya existían en el sistema.

Siguiendo el mismo procedimiento, se procesan las hojas restantes, haciendo uso de los identificadores, limpiando la información, creando nuevas entradas con los datos en caso de que no existan, o almacenando los números de filas si los valores ya existen o presentan errores, y guardando los resultados de estas acciones en cada método.

En el caso de la hoja con las propiedades, se hace uso de la clase que procesa los proyectos como referencia para el proyecto y etapa a los que pertenece cada propiedad. Por esta razón, ambas hojas tienen el mismo identificador "Proyecto + Etapa". De la misma forma, se extrae la información de cada columna y se intenta agregar una nueva entrada en la tabla *Property* en la base de datos. Si existe una propiedad con el mismo código y con en el mismo proyecto y etapa, ésta no es agregada, y eso se indica en los resultados.

La hoja de clientes, en cambio, no requiere referencias de las otras hojas. Se utiliza el rut del cliente como identificador, y también para buscar si hay una entrada con ese campo idéntico en la base de datos. Si no lo hay, entonces se crea una nueva entrada con la información extraída de las columnas.

Finalmente, para la hoja de relaciones, se toman como referencia las clases que procesan las propiedades y los clientes, utilizando sus identificadores. Si no existe una relación entre un cliente y la propiedad indicada en la fila, ésta se agrega a la base de datos.

Una vez que se procesan todas las hojas, se almacenan los resultados obtenidos por cada fila en cada una, se ordenan y se entregan para ser visualizadas en la interfaz, detallando la cantidad de filas que se agregaron, cuáles contenían datos que ya existen en la base de datos y cuáles provocaron errores.

De forma adicional, se agregan tres clases para la carga de hojas individuales. Si Magal desea agregar solo nuevas propiedades, lo puede hacer seleccionando un proyecto y etapa, y utilizando un archivo excel sobre la planilla. Esta clase se creó para que no requiera referencias de otras clases al momento de procesar y agregar propiedades. La otra clase cumple una función similar, y se usa para agregar relaciones entre cliente y propiedad.

Por lo tanto, el proceso de carga masiva solamente funciona como un *create*, es decir, agrega nueva información al sistema, y no reemplaza las entradas existentes. El mismo proceso aplica para procesar las hojas individuales. Si Magal desea modificar campos sobre los datos, deben hacerlo de forma manual. Lo mismo se requiere para desactivar propiedades o eliminar relaciones existentes.

3.7. Cambios al Modelo de Datos

Tras un análisis de los cambios y nuevos módulos que se requerían implementar en el sistema, se determinaron los cambios a realizar al modelo de datos, para que éste funcione de la manera esperada. A continuación se describen dichos cambios y se indica la razón por la cual las modificaciones son necesarias.

Project: Esta tabla almacena la información básica de un edificio. Se agregó el campo *rut_admin*, el cual se utiliza para ingresar el rut (sin puntos ni guión) correspondiente al administrador del edificio. De esta forma, cuando el usuario con este rol rut inicie sesión, podrá realizar reclamos sobre áreas comunes de los edificios, en los que su rut coincida con este campo, a diferencia de un usuario normal que solo puede reportar incidentes sobre sus propiedades.

Client Relation Property: Esta tabla representa la relación entre un cliente y una propiedad. En ella se agregó el campo *deed_date*, el cual corresponde a la fecha en la cual se hace efectiva la entrega de la propiedad al cliente. Esta fecha está presente como columna en la planilla Excel descrita en puntos anteriores, por lo que se agrega en las entradas al momento de procesar las hojas de dicha planilla. Esta es una fecha importante para Magal, pues con ella se determina si un cliente puede realizar reclamos o no a una propiedad, dado que el servicio de post-venta o post-arriendo tiene una fecha de vencimiento.

Property: Almacena los detalles de una propiedad y su relación con los edificios. En esta tabla se agregó el campo *is_active*, el cual indica si una propiedad cuenta con el servicio de garantía (post-venta o post-arriendo) disponible. Magal espera poder desactivar las propiedades de forma manual, después de revisar en un ticket si se venció el plazo de garantía, lo que se calcula utilizando el campo '*deed_date*' descrito en el punto anterior.

Visit: Esta tabla almacena toda la información de una visita, incluyendo estado, fotos, atrasos, y otros datos. En ella se agregó el campo *created_date* que guarda la fecha y momento en que se crea la visita. Este campo se utilizará para el cálculo de estadísticas.

FaultInTicket: Esta tabla permite relacionar un problema a un ticket, y cuenta con un campo para saber si éste fue resuelto o no. En ella se agregaron los siguientes campos: *is_approved* y *solved_date*. El campo *is_approved* permite a Magal aprobar o rechazar

individualmente el problema de un ticket. En la plataforma legada, solo se puede rechazar o aprobar un ticket, pero no sus problemas de forma individual. El campo *solved_date* sirve para guardar la fecha cuando es resuelto el problema, y así éste puede ser usado en el cálculo de estadísticas.

ClientUserInfo: Esta es una nueva tabla que se agrega para almacenar la información proporcionada por un cliente al momento de registrarse. Ésta cuenta con campos para sus datos personales, sus datos de propiedad, entre otros, los cuales se listan a continuación: *name*, *rut*, *phone*, *email*, *is_validated*, *document_file*, *document_file_uploaded*, *data_address*, *data_aptnumber*, *data_tower*, *data_commune*, y *is_project_admin*.

Los campos *name*, *rut*, *phone* y *email* corresponden a la información personal y datos de contacto del cliente. El campo *is_validated* permite saber si el usuario con el rut vinculado está validado, ya sea automáticamente por el sistema o por revisión manual. Los campos *data_address*, *data_aptnumber*, *data_tower* y *data_commune* almacenan la información proporcionada por el usuario sobre su propiedad. El campo *document_file* permite almacenar un archivo, y *document_file_uploaded* es un booleano que indica si un archivo fue subido por el usuario. El campo *is_project_admin* es solicitado por Magal para identificar a los usuarios que cumplen el rol de administrador de edificios.

Ticket: Esta tabla almacena toda la información de un reclamo, los problemas, imágenes y además la información personal del usuario que reporta el problema. En ella se eliminaron los siguientes campos: *sender_name*, *sender_rut*, *sender_phone*, *sender_email*, *is_owner*, *owner_name*, *owner_rut*, *owner_phone*, y *owner_email*.

Tal como se indicó antes, en el sistema legado cualquier persona podía reportar problemas, por lo que debía ingresar de forma manual sus datos, y en caso de que no fuera el propietario, debía ingresar también los datos de éste. La razón por la que se eliminan estos campos es porque ahora el usuario debe estar registrado y conectado para poder realizar reportar problemas. Sus datos personales son proporcionados al momento de registrarse, estos se guardan en la tabla *ClientUserInfo*, y ésta se vincula al ticket creado, por lo que no es necesario que dicha tabla los guarde.

Por otra parte, se agregaron los siguientes campos a esa tabla: *client*, *userclient*, *sentbyowner*, y *user_sender*. El campo *client* vincula la información de un cliente disponible en la base de datos a través de una llave foránea a la tabla *Client*, al igual que el campo *userclient* con la tabla *ClientUserInfo*. Magal considera necesario que un empleado suyo, ya sea un encargado de post-venta, administrador o ingeniero, pueda reportar problemas en una propiedad; esto se hace, a través de una vista aparte, a la cual no puede acceder un usuario cliente. Normalmente el proceso sería que el usuario propietario se registre para realizar reclamos, pero un miembro de Magal debe poder realizar uno en lugar del propietario. De esta forma, si el reclamo fue realizado por personal de Magal, se indica a través del campo *sentbyowner*, el cual se marca falso si fue enviado por un usuario distinto al propietario (es decir, personal de Magal), y en el

campo *user_sender* se vincula al usuario que realizó dicho reclamo. Además, es posible que un usuario no se encuentre registrado en la plataforma cuando personal de Magal reporte un problema en su propiedad. En este caso, el campo *userclient* está vacío y la información del cliente se extrae de la tabla *Client*.

TicketSurvey: Esta tabla vincula una encuesta a un ticket cerrado, con campos para la fecha de creación, y si fue respondida, entre otros. Se agregaron los siguientes campos a esta tabla: *expire_date* y *expired*. El campo *expire_date* corresponde a la fecha en la cual la encuesta deja de ser válida, la cual se calcula automáticamente desde el momento que se crea, 30 días después. El campo *expired* indica si la encuesta ya venció.

VisitSurvey: Esta tabla vincula una encuesta a una visita realizada, con campos similares al de la tabla para encuesta de ticket. Se agregaron los campos *expire_date* y *expired* a esta tabla, los cuales tienen la misma semántica que en el caso anterior.

Con estos cambios al modelo de datos, los nuevos módulos pueden funcionar de forma adecuada, manteniendo la estructura general del sistema legado.

3.8. Validación del Registro de Cliente

El sistema con los nuevos cambios contará con el registro de usuarios, para que estos puedan reportar problemas en sus propiedades. Según lo conversado con Magal, la cuenta del cliente debe pasar por un proceso de validación automática al momento de realizar el registro. Para esto, se solicita al usuario datos que serán utilizados en un proceso de validación. Los datos a utilizar son los siguientes: Rut, Dirección, Número de Departamento, Número de Torre, y Comuna.

Cuando el formulario de registro es enviado, se realiza el proceso de validación en el back-end. Primeramente, el rut del cliente se utiliza para hacer una búsqueda en la tabla “Cliente” de la base de datos. El rut es formateado por el sistema (en caso de ser requerido) para agregarle un guión antes del dígito verificador, y luego realizar la búsqueda. Si encuentra una entrada que coincida con el rut, ésta se guarda para ser utilizada más adelante. Dado que no pueden existir dos clientes con el mismo rut, solo es necesario este dato para comprobar si el usuario existe en la base de datos.

A continuación, se verifica la autenticidad del vínculo del usuario con su propiedad, dados los otros datos entregados. Se utiliza la dirección para hacer una búsqueda sobre la tabla “Etapa”. Acá pueden ocurrir dos casos; el primero es que en la búsqueda se encuentre más de 1 resultado, lo que significa que el edificio asociado tiene más de 1 torre. Las etapas pueden tener la misma dirección y pertenecer a un mismo proyecto, pero sus números de torre son diferentes. En este caso, se verifica si el número de torre proporcionado en el formulario coincide con uno de los resultados, y se almacena la instancia correspondiente en una variable.

El otro caso es que la búsqueda solo encuentre un resultado, y en ese caso se ignora lo que el usuario haya ingresado en el campo de número de torre, y se almacena la instancia. Luego, a partir de la instancia guardada se extrae el proyecto vinculado, y se verifica que la comuna de dicho proyecto sea equivalente a la comuna proporcionada en los datos del formulario.

Luego, utilizando el campo “número de departamento” del formulario, el proyecto y etapa obtenidos, se hace una búsqueda sobre la tabla “Propiedad” para encontrar una entrada con campos que coincidan. Si la búsqueda es exitosa, se guarda la instancia en una variable.

Por último, utilizando el cliente y propiedad obtenidos previamente, se hace una búsqueda sobre la tabla “Relación Cliente Propiedad”, para verificar si la relación existe en el sistema. Solo puede existir una relación para una propiedad, por lo que si se encuentra una entrada usando dicho usuario y propiedad, el usuario será validado. Si alguna de las búsquedas previas falló a causa de que los datos no coinciden, o a que la relación no existe, el usuario no será validado.

Una vez completados los pasos anteriores para la validación, se crea el usuario en el sistema con el rol de “Cliente”, y se crea una entrada en la tabla UserClientData para almacenar todos los datos entregados en el formulario, con el campo “is_validated” como verdadero si la validación fue un éxito. La contraseña creada para el usuario es generada de forma aleatoria y enviada al correo proporcionado en el formulario.

3.9. Resumen

En este capítulo se presentó en detalle el diseño de la solución que se desarrolló en este trabajo de memoria, a modo de cumplir con los objetivos propuestos. Se presentaron bosquejos de las interfaces de cada nuevo módulo, y se explicaron los procesos que manipulan y procesan los datos. En los próximos capítulos se indican los cambios realizados durante la implementación, con respecto a lo presentado en este capítulo.

Capítulo 4

Implementación

Tomando en consideración el diseño y decisiones presentadas en los capítulos anteriores, se desarrollaron los módulos descritos, a modo de cumplir con los objetivos propuestos. Se trabajó primero en el módulo que permite la carga masiva de datos, a continuación en el módulo que se divide en un formulario para registro de usuarios y vistas, para su ingreso al sistema. Por último, se implementó el tercer módulo, que corresponde a un panel de estadísticas del sistema.

Durante el trabajo se llevó a cabo un proceso de desarrollo iterativo, donde se trabajó en cada módulo de manera individual, y a su vez, para cada módulo se trabajó en paralelo el desarrollo de backend y las interfaces correspondientes. Se mantuvo contacto permanente con el personal de Magal, para la resolución de dudas en torno a casos borde, y comprobar que los diseños y desarrollo de las interfaces sean los correctos y acordes a lo que ellos esperaban. Para ello, se realizaron reuniones puntuales para mostrar los avances y recibir retroalimentación para corregir lo que sea necesario. A continuación se describe en secciones el detalle de cada módulo implementado, con sus interfaces y funcionalidades correspondientes.

4.1. Módulos de Carga de Datos

En esta sección se revisa en detalle el módulo y vistas que comprenden el proceso de carga de datos. Primeramente, se tiene en cuenta el archivo Excel que utiliza la planilla mencionada en las secciones previas, la cual tiene un formato específico y es utilizada como template para la carga de información al sistema.

De acuerdo a lo presentado en el punto 3.3, se eliminó el recuadro que agrega nuevos proyectos, ubicado en la pestaña que administra los proyectos actuales. Además, se agregó una nueva pestaña a la vista de administración, en la cual se adicionaron 3 recuadros para la carga de datos (Figura 4.1). Esta pestaña se encuentra en una vista sólo visible para usuarios administradores. Se agregó un botón que permite descargar el *template* del archivo Excel directamente desde esta vista, dado que fue solicitado por Magal.

En el primer recuadro se muestra un texto descriptivo sobre la función que cumple la carga del archivo y un botón para ejecutar la acción. Al realizar la carga con este recuadro, se procesan las 4 hojas definidas en la planilla, es decir, los proyectos, las propiedades, los clientes y las relaciones entre cliente y propiedad.



Figura 4.1: Interfaz de la carga de datos en el sistema.

El segundo recuadro cumple una función similar, también permitiendo al usuario subir un archivo Excel para realizar carga de datos, pero a diferencia del primero, este solo procesará la hoja de los clientes. El tercer recuadro, funciona igual que el anterior, pero éste sólo procesa la hoja que contiene las relaciones de clientes con las propiedades.

Se hizo esta división dado que el flujo normal de Magal para agregar un nuevo edificio a la base de datos, consiste en primero crear el proyecto y etapas con sus datos básicos, dirección, comuna, y torres. Se cuenta con una fecha de recepción municipal que puede no estar disponible en el momento en el que se realiza la carga, por lo que ésta es modificable de forma manual. Luego, Magal agrega cada cierto período de tiempo nuevos clientes a la base de datos, y cuando hacen entrega de las viviendas, agregan las relaciones con dichos clientes a sus propiedades correspondientes.

Después de seleccionar un archivo y realizar la carga usando el botón de Carga, éste se procesa, y en la interfaz se muestran los resultados de la carga. En la Figura 4.2 se muestra el primer recuadro de carga, después de recibir los resultados del procesamiento. Por cada hoja se muestra la cantidad de filas agregadas al sistema, la cantidad de filas existentes y la cantidad de filas que produjeron un error al ser procesadas. Tanto para las filas existentes como para las que dieron errores, se puede ver el detalle donde se indica el número de fila y el mensaje correspondiente. Así mismo, al terminar el procesamiento en el segundo y tercer recuadro, se despliegan los resultados, pero en ambos casos solo se muestran los resultados de una hoja.

Resultados de la Carga

Hoja Proyectos
 Agregados : 0
 Existentes : 9

Detalle de Proyectos Existentes ▼

Con Error : 0

Hoja Propiedades
 Agregadas : 719
 Existentes : 1695

Detalle de Propiedades Existentes ▼

Con Error : 6

Detalle de Propiedades con Error ▲

Fila 212: Código vacío

Fila 348: Código vacío

Fila 494: Código vacío

Fila 547: Código vacío

Fila 1141: Código vacío

Figura 4.2: Enfoque a los resultados de la carga de datos.

En esta pestaña, y haciendo uso de la planilla Excel definida en capítulos anteriores, Magal puede realizar la carga de datos. Tomando en cuenta que existe información de un proyecto que Magal necesita configurar después de agregarlo a la base de datos, se modificó la vista existente que lista los proyectos, en específico el recuadro que muestra el detalle de un proyecto seleccionado, de forma que se puedan editar los campos relevantes.

En la Figura 4.3 se muestra el recuadro que aparece al seleccionar un proyecto, donde se pueden editar sus campos. Como se mencionó antes, Magal considera necesario poder editar la fecha de recepción municipal de forma manual, en el caso de que no cuenten con ella durante la carga de datos, por lo que en este recuadro se permite cambiar dicha fecha. Dado que puede haber usuarios de tipo “Cliente” a cargo de uno o más proyectos, bajo el rol de “Administrador de Edificio”, se configuró esta vista para que se pueda ingresar de forma manual el rut del cliente que está a cargo. Este campo se usa para que el usuario con ese rol y rut pueda reportar problemas sobre áreas comunes.

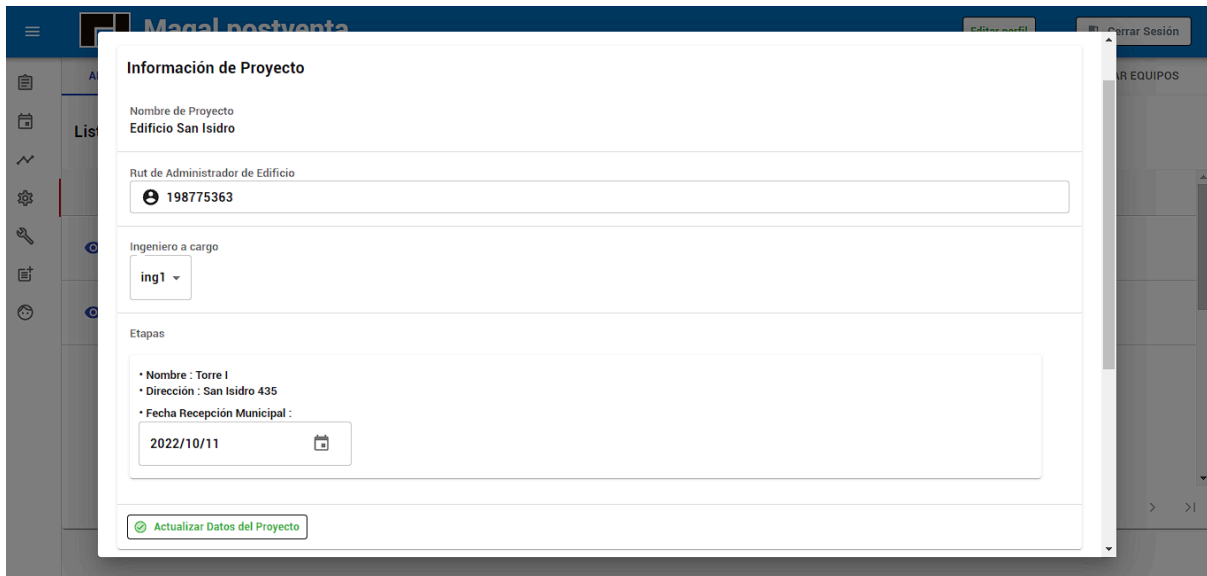


Figura 4.3: Cuadro emergente para editar la información de un proyecto.

Además, se configuró esta vista para que también se puedan ver y editar de forma manual cada una de las propiedades asociadas al proyecto. En la Figura 4.4 se listan las opciones adicionales sobre el recuadro de edición de un proyecto. Estas acciones permiten el manejo de las propiedades existentes en un proyecto, la carga de nuevas propiedades y la opción de desactivar el proyecto, en el caso de que Magal considere que el servicio de garantía ya no deba realizarse en él.

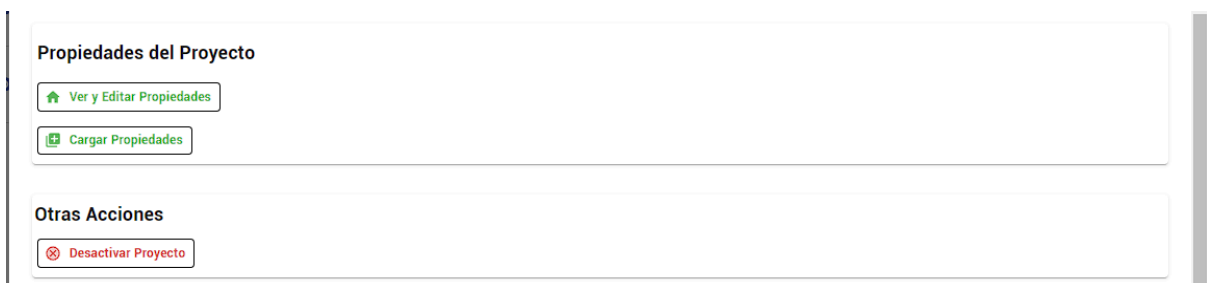


Figura 4.4: Acciones adicionales en el recuadro de edición de proyecto.

Al seleccionar el botón “Ver y Editar Propiedades”, se abre un recuadro con una tabla que lista todas las propiedades. En la Figura 4.5 se muestra la tabla con las propiedades del proyecto seleccionado, indicando el código, el estado de la propiedad (considerando que el plazo de garantía tiene vencimiento para la vivienda, este servicio puede quedar desactivado), el tipo de propiedad (departamento, bodega, etc.) y la torre en la cual está ubicada. Se incluyó además un filtro para buscar las propiedades por código y tipo.

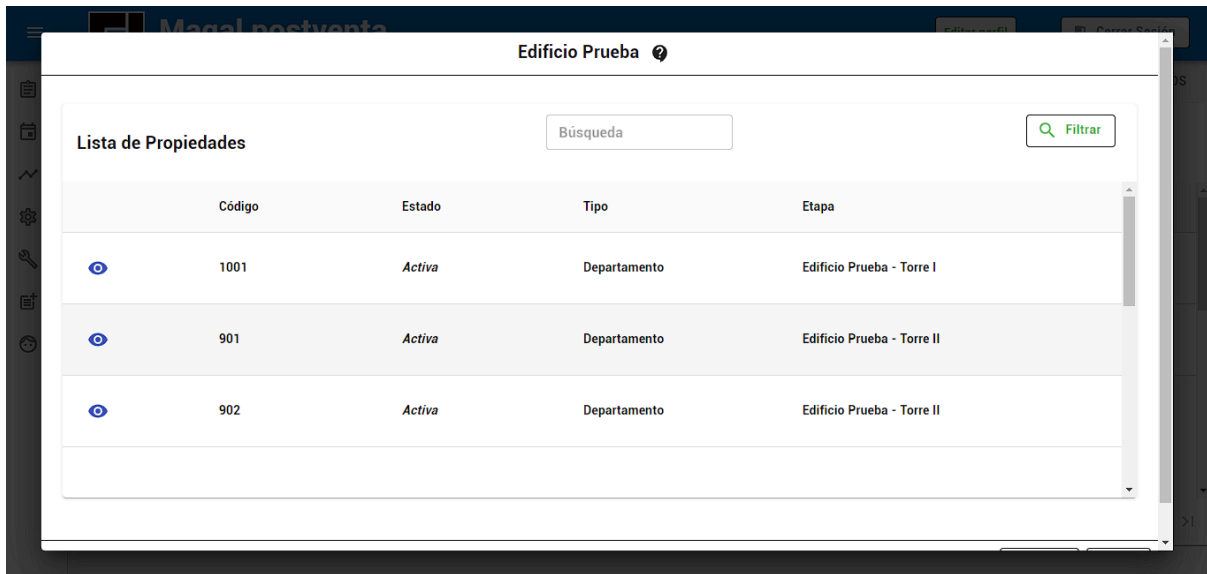


Figura 4.5: Listado de propiedades de un proyecto seleccionado.

Al hacer clic en el icono de ojo ubicado al inicio de una fila, se abre un nuevo recuadro con el detalle de la propiedad seleccionada. En este recuadro se pueden ver y editar los datos, entre ellos el código, el tipo, y la fecha de entrega al cliente con el cual posee una relación. En la Figura 4.6 se muestra un enfoque al recuadro de detalle de una propiedad que posee una relación con un cliente. Esta relación se puede eliminar, según se considere necesario. También se puede editar la fecha de entrega, en caso de un error de tipeo al momento de cargar los datos.

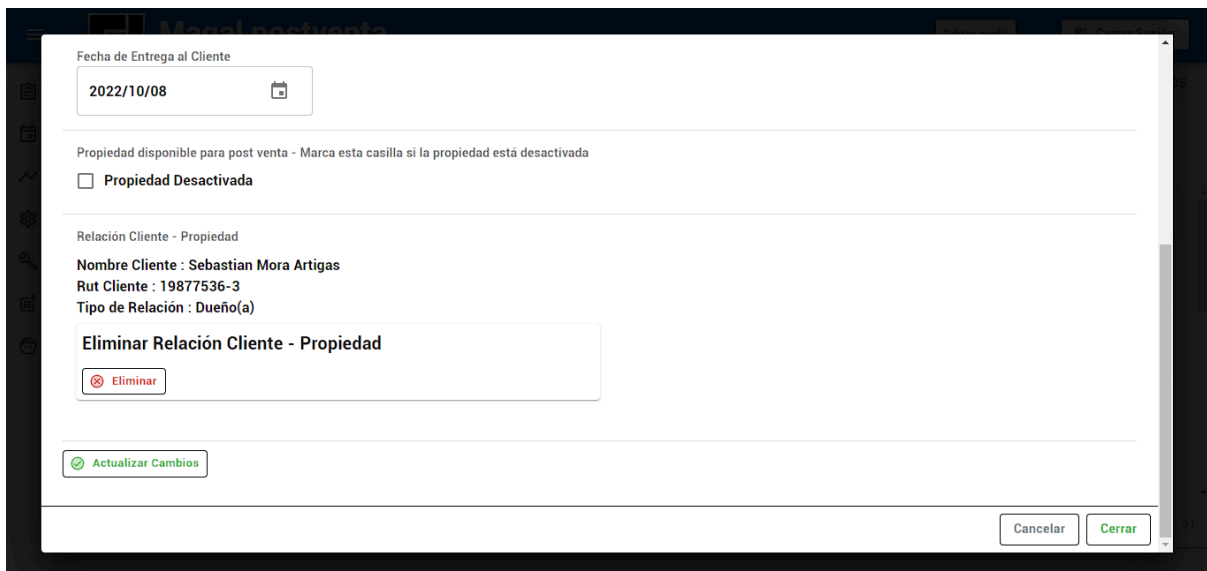


Figura 4.6: Enfoque al recuadro de detalle de una propiedad seleccionada.

Por otro lado, si el usuario presiona el botón “Cargar Propiedades” (mostrado en la Fig. 4.4) se abre un nuevo recuadro emergente con una estructura similar a los recuadros de carga masiva. En este recuadro (Fig. 4.7) se puede hacer la carga de un archivo Excel que cumpla

con la estructura correcta, a fin de cargar nuevas propiedades al proyecto seleccionado. Este archivo es una adaptación de la planilla con las 4 hojas originalmente definidas, donde solo se dejó la hoja de propiedades y se quitó la columna que sirve como identificador. Además, es posible descargar el template del archivo directamente desde este recuadro. El usuario debe seleccionar la torre correspondiente y subir el archivo para poder realizar la carga. Los resultados se despliegan en el mismo, y siguiendo el mismo formato que en la carga masiva de datos.



Figura 4.7: Enfoque al recuadro de carga de propiedades.

Con las nuevas vistas, pestaña y recuadros descritos anteriormente, Magal cuenta con un mayor manejo de los datos sobre sus edificios y propiedades, siendo posible agregar datos de forma masiva, editar y eliminar de forma manual aquellos que presenten errores o según consideren necesario.

4.2. Módulo de Registro de Clientes

En esta sección se detallan las vistas correspondientes al módulo de registro de clientes, su acceso al sistema, interacciones, y el manejo por parte de Magal sobre estos. Estos se explican en las siguientes subsecciones.

4.2.1 Interfaces del Usuario Cliente

Primeramente, se describirán los cambios a la página principal, visible para un usuario que no se ha logueado al sistema. La vista principal del sistema recibe al usuario con el banner de Magal, un botón superior para iniciar sesión y un botón a mitad de página que permite al usuario reportar problemas. Previamente, el botón de reporte abría un formulario en el cual el usuario debe seleccionar su propiedad, ingresar sus datos personales de forma manual, seleccionar los problemas y subir imágenes para crear un ticket (reclamo).

Como se ve en la Figura 4.8, se hicieron cambios para que este botón no muestre el formulario de reclamo, sino que solicita al usuario ingresar al sistema para realizar las

solicitudes, con la posibilidad de registrarse si no posee una cuenta en la plataforma. A su vez, si el usuario está registrado pero olvidó su contraseña, clickeando el texto marcado en rojo puede ir a una vista que le permitirá restaurar su contraseña utilizando su correo.

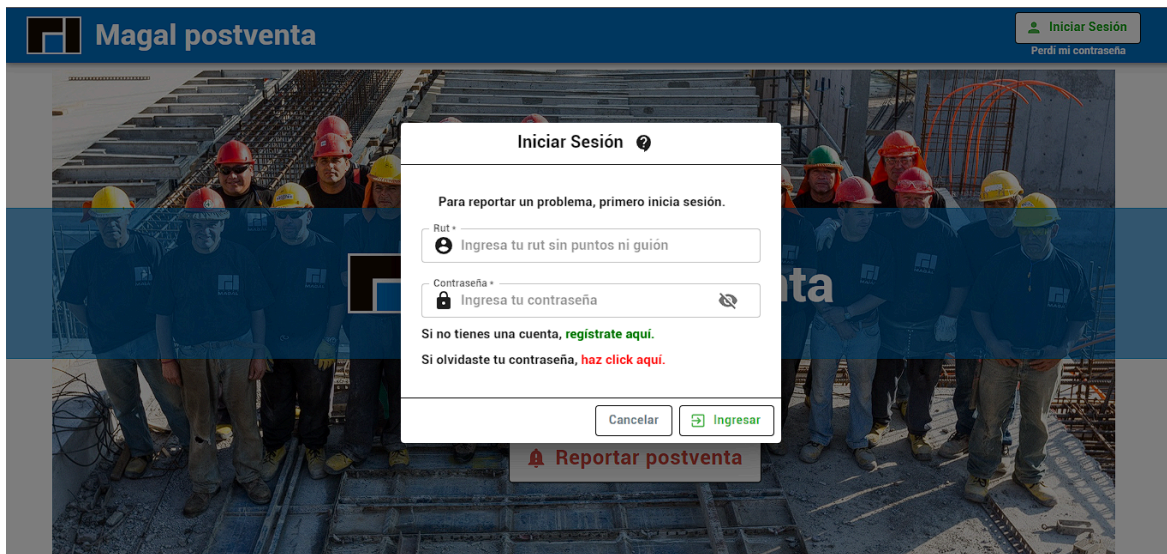


Figura 4.8: Recuadro que solicita inicio de sesión al usuario.

Al hacer clic en el texto marcado en verde que dice “regístrate aquí” se abre un nuevo recuadro (Fig. 4.9) emergente con el formulario de registro. Acá se solicitan los datos personales del usuario, incluyendo rut e información de contacto. El rut ingresado es validado en el mismo formulario haciendo un cálculo con el dígito verificador, es decir, el último dígito ingresado. Se solicita ingresar los datos de su propiedad, o de una de sus propiedades en caso de que posea más de una.

Crear nueva cuenta ?

Completa los datos indicados a continuación para crear una nueva cuenta.

Rut *
Ingresar tu rut sin puntos ni guión

Nombre Completo *
Nombres y Apellidos

Email *
Ingresar tu correo electrónico

Teléfono *
+569

Ingresar los datos de tu propiedad. Si posees más de una propiedad, ingresa los datos de una de ellas.

Dirección
Ejemplo : Las Hualtatas 5099

Número Depto
Ingresar el número de tu departamento

Torre
I

Comuna
Ingresar tu comuna

Cancelar Registrarse

Figura 4.9: Enfoque al recuadro de formulario de registro.

Cuando el usuario completa todos los campos y el rut no presenta errores, se habilita el botón para proceder con el registro. Esto desencadena el proceso de validación en el back-end y se crea el usuario utilizando el rut del cliente. En caso de que ya exista una cuenta en el sistema con el rut ingresado, el registro se cancela y se indica al usuario con un mensaje de advertencia.

En el caso de que no exista un usuario con el rut, se procesan los datos ingresados para determinar si el usuario queda como cliente validado en el sistema o no. Luego, se crea la cuenta y se muestra un mensaje emergente, indicando que el proceso fue un éxito y la contraseña fue enviada al correo proporcionado.

La solución propuesta en el capítulo anterior para este módulo sufrió cambios al momento de desarrollarla. Magal estableció que un usuario cliente que no está validado, debe proporcionar un documento que acredite que es el propietario de la vivienda que ingresó en el formulario, para así facilitar a Magal el trabajo de validación manual. Así, la página que ve un usuario cliente que inicia sesión se dividió en diversas pestañas (Fig. 4.10).



Figura 4.10: Vista principal de usuario cliente que inicia sesión.

La primera pestaña de esta interfaz cuenta con un botón que abre el formulario de reclamos, y si el usuario no está validado, se indica con un mensaje y se le da la opción de subir un archivo. Este archivo puede ser una imagen o archivo *.pdf*, el cual será luego revisado por el personal de Magal, a fin de validar de forma manual su cuenta. De todas formas, el usuario puede reportar problemas en las propiedades que tengan una relación existente en la base de datos con su rut.

En la Figura 4.11, se muestra el primer paso del formulario de para reportar problemas, el cual se abre tras presionar el botón “Enviar Requerimiento” (Fig. 4.10). A diferencia del

sistema legado, el usuario ahora solo puede reportar sobre sus propiedades, las cuales son obtenidas utilizando su rut y verificando las relaciones existentes en la base de datos.

Si el usuario posee el rol de “administrador de edificio”, el sistema realiza una búsqueda adicional para encontrar los proyectos en que el campo “*rut_admin*” coincida con el rut del usuario. Estos proyectos se agregan a las opciones disponibles, para que el usuario también pueda reportar sobre áreas comunes.

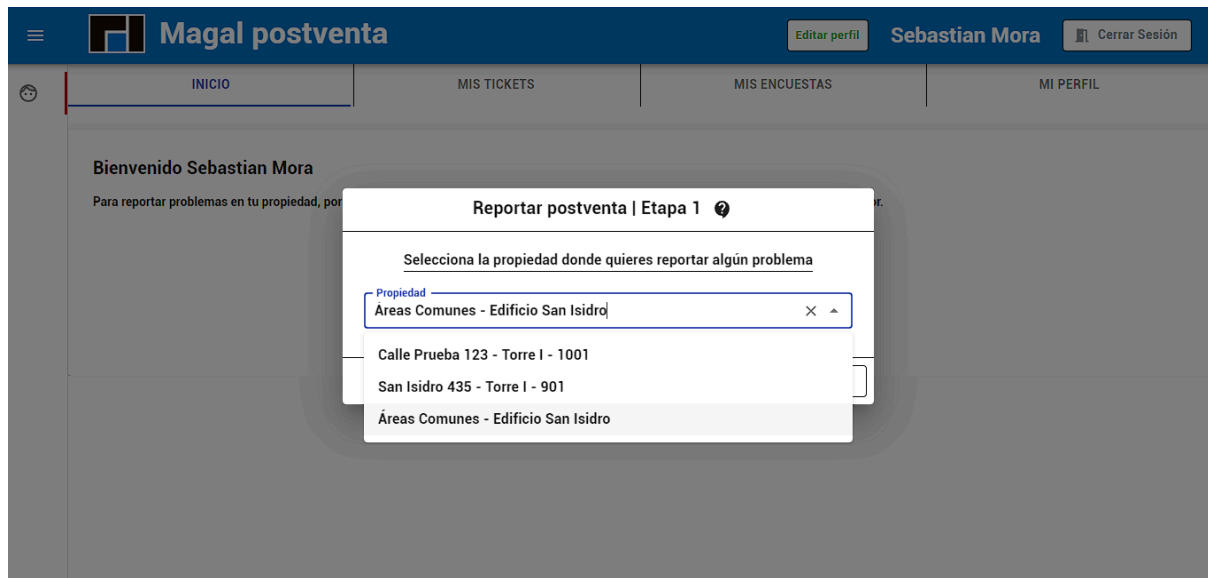


Figura 4.11: Primer paso del formulario de reclamos.

En el formulario, el usuario primero debe seleccionar una propiedad de entre las opciones para continuar. Los siguientes pasos son los mismos que tenía el sistema legado, a excepción del paso que solicita los datos personales de quien reporta, dado que el usuario ya existe en el sistema. Se deben especificar los problemas, incluir imágenes y una descripción para enviar el reclamo.

En caso de que el usuario posea más propiedades, y éstas no estén disponibles entre las opciones, dicho usuario se debe comunicar con Magal para que las relaciones sean agregadas a la base de datos y así poder reportar problemas.

En el sistema legado, un cliente solo puede saber del estado de sus reclamos a través de una página donde debe ingresar su rut y el ID del ticket, el cual es proporcionado a través de un correo el momento en que es creado. Esto solo permitía al usuario ver el estado de reclamos uno a uno y de forma manual.

En el nuevo sistema, el cliente que inicia sesión puede ver sus tickets (reclamos) en la pestaña “Mis tickets” (Fig. 4.12). Esta pestaña cuenta con una tabla que lista todos los tickets asociados al rut del usuario. Se pueden dar casos en que personal de Magal realiza un reclamo a una propiedad, la cual su propietario no está registrado en el sistema. Por esta

razón, se hace una búsqueda sobre la tabla “*Client*” en vez de la tabla “*UserClientData*” para encontrar la instancia que coincida con el rut, y a continuación se obtienen todos los tickets donde el cliente asociado sea el encontrado.

Propiedad	Fecha reporte	Problema	Estado
Calle Prueba 123 - Torre I - Departamento 1001	08/10/23 16:37	• Acceso - Cielo - Estetico	Cerrado
Calle Prueba 123 - Torre I - Departamento 1001	12/10/23 12:28	• Acceso - Cielo - Estetico	Rechazado
Edificio San Isidro	19/10/23 01:57	• Acceso / Hall acceso - Baranda - Funcionamiento	Por aprobar

Figura 4.12: Pestaña que lista los reclamos del usuario cliente.

Si el usuario selecciona un reclamo haciendo clic en el icono de ojo ubicado a la izquierda de la fila, se abre un recuadro que muestra la información proporcionada en éste, el estado y las visitas realizadas. De esta forma, el usuario cuenta con un acceso rápido al detalle de sus reclamos, a diferencia del sistema legado.

La siguiente pestaña lista todas las encuestas generadas sobre los tickets del usuario y las visitas agendadas. En el sistema legado, el cliente solo podía acceder a las encuestas a través de un enlace enviado por correo, y cada una tenía un enlace único. Estas tenían una duración indefinida, por lo que permanecían abiertas hasta que el usuario decidiera contestarlas. En el nuevo sistema, todas las encuestas asociadas a tickets y visitas de un usuario se listan en una tabla, como muestra la Figura 4.13.

Tipo de Encuesta	Estado	Fecha creada
Encuesta de Ticket	Sin contestar	12/10/23 12:28
Encuesta de Visita	Sin contestar	12/10/23 12:27

Figura 4.13: Pestaña que lista las encuestas del usuario cliente.

En cada fila de la tabla se indica el tipo, su estado y la fecha de creación. Tal como se puede ver en la Figura 4.13, cada fila posee un icono en su primera columna. Si el usuario clickea sobre un icono, se abre una nueva pestaña para poder contestar la encuesta correspondiente a esa fila. En el sistema legado, el cliente solo podía acceder a las encuestas una a una a través de enlaces, y a causa de esto, no tenía manera de saber si ya respondió una encuesta excepto revisándola con el enlace.

Como se indicó antes, en el sistema legado las encuestas tenían una duración indefinida. Según lo solicitado por Magal y los cambios realizados al modelo de datos, las encuestas ahora tienen un plazo de 30 días para ser contestadas. Si el usuario no contesta una encuesta durante este período, al transcurrir el plazo establecido, la encuesta se marcará como vencida. Esta revisión se hace para todas las encuestas de forma automática con una tarea de cron diaria, aunque también se ejecuta sobre una encuesta individual cuando el cliente abre su enlace, a fin de enviar las preguntas a la interfaz correspondiente.

Por último, la siguiente pestaña disponible para el usuario, le permite a éste revisar su cuenta y realizar cambios sobre sus datos de contacto. En la Figura 4.14 se muestra la pestaña de un usuario cliente que revisa su perfil, en la cual puede editar su correo electrónico y número de celular. En la misma pestaña hay un recuadro ubicado abajo donde se puede editar la contraseña. El usuario puede acceder a esta misma pestaña si clickea el botón “Editar Perfil” ubicado en la esquina superior derecha de la página.

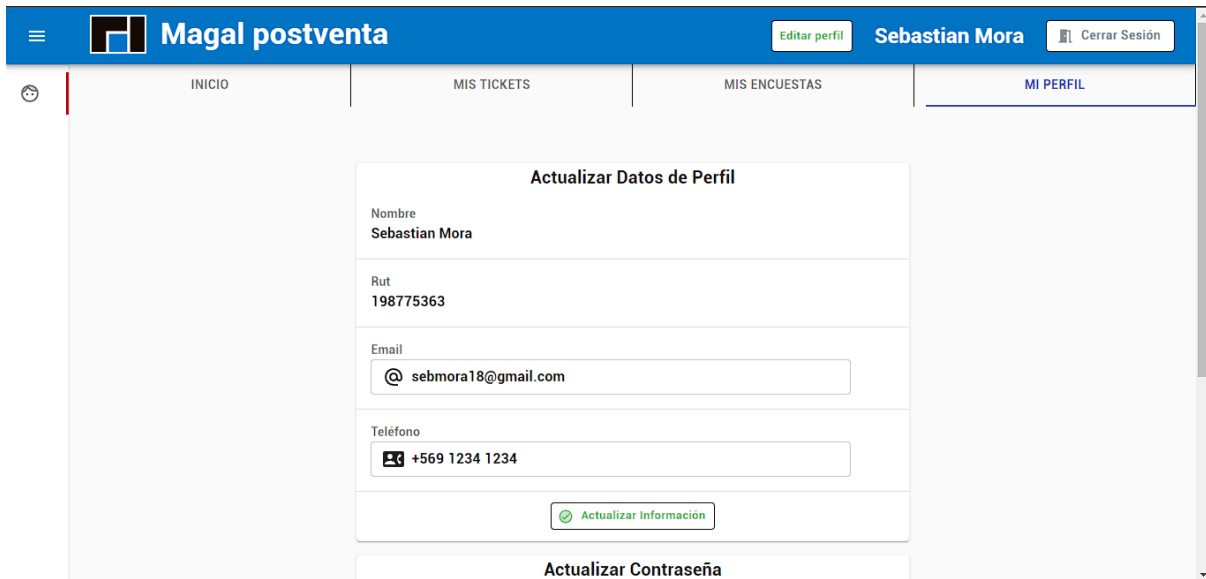


Figura 4.14: Pestaña que permite al cliente editar su perfil.

4.2.2 Vistas de Administración

Sumado a los cambios y las vistas agregadas para los nuevos usuarios que se registran, se realizaron ajustes sobre vistas existentes y se crearon nuevas vistas. En su mayoría, la nueva funcionalidad es para usuarios con rol de administrador, por ejemplo, para permitirle a Magal aprobar aquellos usuarios que no fueron validados por el proceso automático al momento del registro.

Primeramente, dados los cambios hechos al modelo de datos y la nueva estructura de los reclamos, se actualizaron todas las interfaces que muestran información de estos, y se ajustaron los *serializers* que proveen la data desde el back-end. Todas las vistas y *serializers* que utilizan los datos de contacto del cliente, los cuales estaban directamente vinculados al reclamo en el sistema legado, ahora los reciben utilizando el vínculo con la tabla “*UserClientData*” o en su defecto, la tabla “*Client*”.

En el sistema legado, personal de Magal debía revisar y verificar de forma manual la información proporcionada en el reporte de problemas, a fin de tener certeza de que el usuario que reporta es el propietario, y los datos de su propiedad son los correctos. Esto provocaba retrasos tanto en la aprobación de los reclamos, como en el agendamiento de visitas, lo cual perjudicaba aquellos reclamos con un grado de urgencia elevado.

A diferencia del sistema legado, el nuevo sistema solo permite el reporte de problemas a usuarios que se encuentren registrados y conectados. De esta forma, la aprobación de los tickets es más sencilla. Si el usuario que reporta está validado en el sistema, personal de Magal ya no necesita verificar los datos personales y de contacto, ya que el usuario solo puede hacer un reclamo en una propiedad si cuenta con una relación existente en la base de datos. Si el usuario no está validado, Magal hará una revisión adicional al momento de ver los reclamos.

En la Figura 4.15 se muestra el recuadro con los detalles de un reclamo. En éste se indican los datos personales del cliente, si está validado en el sistema, y si fue el propio cliente quien hizo el reclamo. De todas formas, el usuario cliente no puede realizar reclamos asociados a una propiedad, si la relación entre ésta y su rut no existe en la base de datos.

Información de ticket

Datos Cliente

Nombre: Sebastian Mora Artigas

Rut: 19877536-3

Email: sebmora18@gmail.com

Teléfono: +569 1234 1234

El Cliente está validado. El Requerimiento fue enviado por el propietario.

Figura 4.15 : Enfoque al recuadro de reclamo con los datos del cliente.

Por otro lado, se agregó una nueva vista que conserva el formulario de reclamos del sistema legado. La diferencia es que éste sólo es visible para personal de Magal, y permite realizar reclamos sobre propiedades, sin importar si el propietario no está registrado en el sistema. Para esto se incluyó un icono en la barra lateral con los accesos rápidos, tal como se ve en la Figura 4.16. Al clickear el botón “Enviar requerimiento”, se abre un recuadro con el formulario.

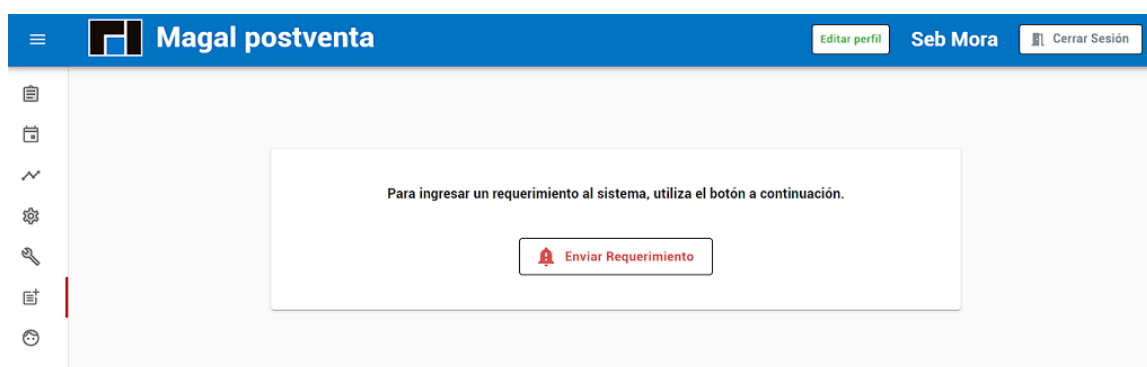


Figura 4.16: Vista que permite a Magal realizar reclamos.

El formulario original consistía en cuatro pasos, pero se ajustó para que sólo sean tres. El primer paso permite seleccionar el proyecto, etapa, tipo y propiedad. El segundo paso, que originalmente solicitaba al usuario ingresar sus datos personales y de contactos, fue removido dado que el usuario que reporta ya está conectado en el sistema. Se conservaron los pasos tres y cuatro, que solicitan especificar los problemas y adjuntar las imágenes correspondientes.

Cuando el reclamo es realizado por un usuario que no es el propietario, al momento de revisar el detalle, se indica en la información del cliente (Figura 4.17). En este caso, se indica que el cliente no está validado en el sistema, y cuál es el usuario (personal de Magal) que reportó los problemas de la propiedad.

Información de ticket

Datos Cliente
Nombre: Usuario Prueba 4
Rut: 18285721-1
Email: sebmora18@gmail.com
Teléfono: +569 15721273
El Cliente no está validado.El Requerimiento fue enviado por el usuario sebmora

Figura 4.17: Enfoque al recuadro de reclamo realizado por un usuario distinto al propietario.

Considerando que el nuevo sistema cuenta con un registro de usuarios, los cuales se intentan validar de forma automática cuando crean sus cuentas, existirán casos en que esta validación falla, ya sea por errores de información al ingresar la propiedad, porque el cliente no existe en la base de datos, o porque la relación con su propiedad no existe en la base de datos.

A modo de revisar los clientes que no fueron validados, se agregó una nueva pestaña en la vista de administración. A solicitud de Magal, se creó una nueva vista para reubicar pestañas, ya que la vista de administración tenía muchas y sería abrumador para el usuario tener tantas opciones. Se movieron 3 pestañas a una nueva vista que se puede acceder a través de un nuevo icono agregado en la barra lateral izquierda. Luego, se agregaron 2 pestañas nuevas en la vista de administración, una para validar clientes, y otra para administrar usuarios tipo cliente.

Al ingresar a la pestaña “Validar Clientes” en la vista de administración (Fig. 4.18), se listan todos los usuarios de tipo “Cliente” que no están validados. Por cada fila se muestra el rut, el nombre del cliente y un indicador para saber si subió un documento que acredite ser el propietario; en este caso, de la propiedad proporcionada en el formulario. Cada fila cuenta con un icono de un ojo, y al hacer clic en este icono, se abre un recuadro con los detalles del cliente.

Rut	Nombre	Documento
237 [redacted]	Francisco [redacted]	Sin subir
251 [redacted]	Sebastian [redacted]	Sin subir

Figura 4.18: Pestaña de administración que lista los clientes no validados.

El sistema realiza una búsqueda sobre la tabla “*UserClientData*” para obtener todas las entradas que tengan el campo “*validated*” como falso. Cada una pasa por un *serializer* que analiza los datos proporcionados en el formulario (los cuales están guardados en la tabla), con el fin de encontrar las coincidencias en las tablas “Propiedad”, “Cliente” y “Relación Cliente-Propiedad”. Por ejemplo, si el cliente no existe en la base de datos (al realizar la búsqueda del rut en la tabla Cliente), se almacena un mensaje que lo indica. Se verifica además el formato del archivo subido, si es una imagen se entrega formateada para que se pueda visualizar en la interfaz, y si es un archivo *.pdf* se muestra el nombre con la opción de descargarlo.



Figura 4.19: Enfoque al recuadro emergente tras seleccionar cliente no validado.

En el recuadro con el detalle se presentan todos los datos que el usuario proporcionó en el formulario (Fig. 4.19). Si el cliente proporcionó un documento, en la celda “Documento proporcionado” se muestra el nombre del archivo si es en formato *.pdf*, el cual se puede clicar y esta acción desencadena la descarga del archivo. En cambio, si es una imagen, ésta se muestra como una miniatura en la misma celda y se puede visualizar en un recuadro emergente con su tamaño completo.

En la celda contigua se indican los motivos por los que falló la validación. Esto permite a Magal tener un mayor entendimiento de la situación con el cliente actual. Se muestran además los datos de la propiedad proporcionados. Se agregó una celda adicional que muestra todas las propiedades que cuentan con una relación con el rut del cliente, que se obtienen con una búsqueda sobre la tabla “Relación Cliente Propiedad”.

El usuario cliente se puede asignar como administrador de edificio marcando la casilla correspondiente. Si el administrador decide validar el usuario, debe hacer clic en el botón “Siguiente” ubicado abajo de las casillas con datos. Esto despliega un panel adicional con una acción requerida para continuar con la validación.

En este panel (Fig. 4.20) se debe seleccionar una propiedad, la cual quedará vinculada al usuario cliente. Si los datos de la propiedad proporcionados en el formulario coinciden con entradas en la base de datos, se seleccionan de forma automática el proyecto, torre y número de propiedad según corresponda. Si, por ejemplo, el cliente ingresó de forma correcta la dirección y comuna, el proyecto queda seleccionado de forma automática, mientras que la torre y el número debe seleccionarlos el administrador en este panel.

Asignar administrador

Asignar usuario como administrador de edificio

• Se creará nuevo cliente en la base de datos.

Proyecto	Torre	Número
Calle Prueba 123	Torre II	901

• La propiedad no tiene una relación previa. Se creará nueva relación con el cliente.

Figura 4.20: Enfoque al paso requerido para validar a un usuario cliente.

En este mismo panel se indica si los datos del cliente ya existen en el sistema, o si esta acción creará una nueva entrada en la base de datos. También se indica si la propiedad seleccionada cuenta con una relación previa o no, ya que al confirmar la validación, se eliminará la relación existente (en caso de que la tenga) y se crea una nueva relación con el cliente actual. Al hacer clic en el botón que confirma, el cliente queda validado, se agrega en la base de datos si corresponde. Además, se crea una nueva relación en el sistema con la propiedad seleccionada.

Por otro lado, en la pestaña “Administrar Clientes” (Fig. 4.21) se listan todos los usuarios tipo clientes registrados en el sistema. Magal considera necesario ver de forma individual, por ejemplo, cuántos reclamos ha hecho un cliente, entre otros datos. Se agregó también un filtro en esta tabla para facilitar a Magal la búsqueda de usuarios, considerando que la lista crecerá con el tiempo.

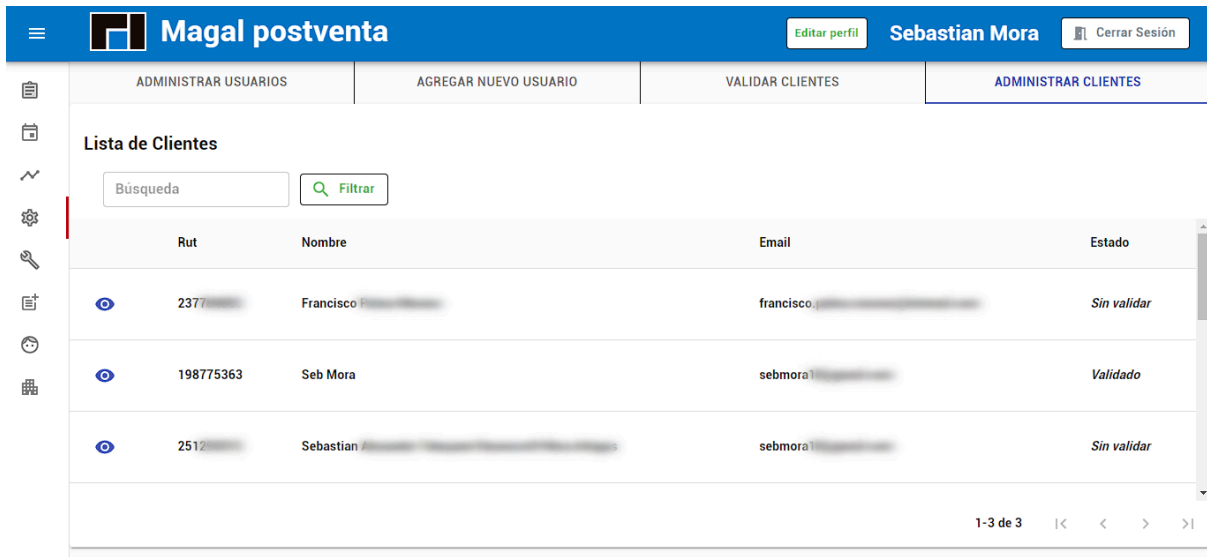


Figura 4.21: Pestaña para administrar clientes registrados en el sistema.

Finalmente, si un usuario de Magal considera necesario ver los reclamos que ha hecho sobre propiedades (en lugar del propietario), puede hacerlo seleccionando el icono que representa a un usuario en la barra lateral izquierda. En esta vista se cuenta con dos pestañas, una de inicio que da la bienvenida al usuario, y la otra que lista los reclamos. Al seleccionar la pestaña “Mis Tickets” (Fig. 4.22) el usuario puede ver todos los tickets que ha creado sobre otras propiedades.

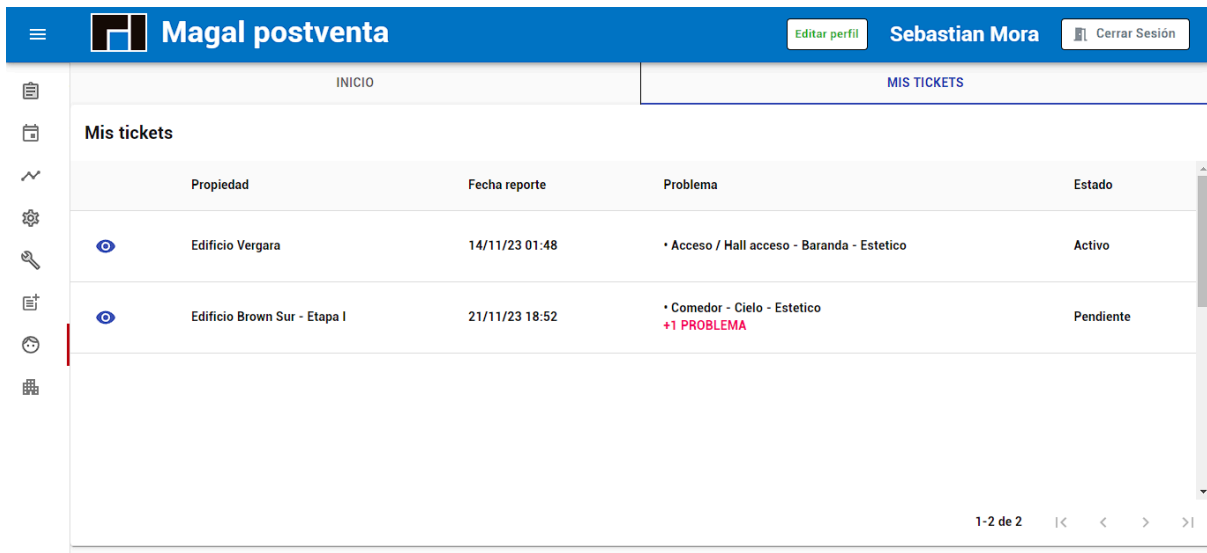


Figura 4.22: Vista con los reclamos hechos por usuario de Magal

Si en la vista mostrada en la Figura 4.22 el usuario selecciona uno de los reclamos de la lista, entonces puede ver la información con el mismo nivel de detalle que al seleccionar uno en la página de tickets. A solicitud de Magal esta vista no permite realizar acciones sobre el ticket seleccionado.

4.3. Panel de Estadísticas

A continuación se detalla la implementación realizada para adicionar el panel de estadísticas descrito en los objetivos de esta memoria. Para esto, se creó una nueva vista con distintos recuadros, la cual permite desplegar y visualizar las distintas opciones disponibles. En esta vista, se agregó un método que recibe información del back-end, el cual es ejecutado de forma automática al abrir la vista por primera vez. Un usuario autorizado puede también invocar manualmente este método del backend, por ejemplo, para cambiar los parámetros y visualizar información diferente.

Primeramente, se agregó un recuadro donde el usuario puede seleccionar los datos a analizar a través de un selector de opción múltiple, y dos botones para seleccionar fechas, que permiten seleccionar el período durante el cual se desean visualizar los datos. Las opciones del selector múltiple permiten ver los datos descritos en la sección 1.3. Además, se agregaron 3 opciones adicionales para ver detalles sobre los tickets, visitas y encuestas.

En la Figura 4.23 se muestra la vista principal que se agregó para las estadísticas, en la cual el usuario ve el recuadro con el selector de datos, y los botones de fechas para seleccionar el período de visualización de los datos. Se agregó además un recuadro adicional, en el cual se muestra un resumen de los tickets, visitas y encuestas generadas en el período seleccionado.

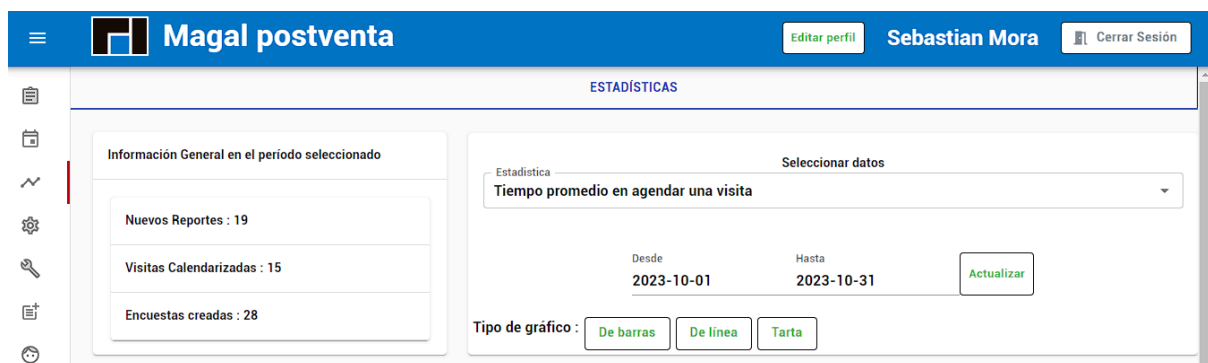


Figura 4.23: Recuadros de la vista principal de estadísticas.

A partir de los datos obtenidos, en esta misma vista se genera un gráfico debajo de los recuadros, el cual por defecto es un gráfico de barras (Figura 4.24). Para esto se utilizó la librería *Recharts* descrita en el capítulo 2. Además, se agregaron 2 alternativas adicionales de gráficos para visualizar los datos, las cuales se pueden alternar seleccionando los botones correspondientes.

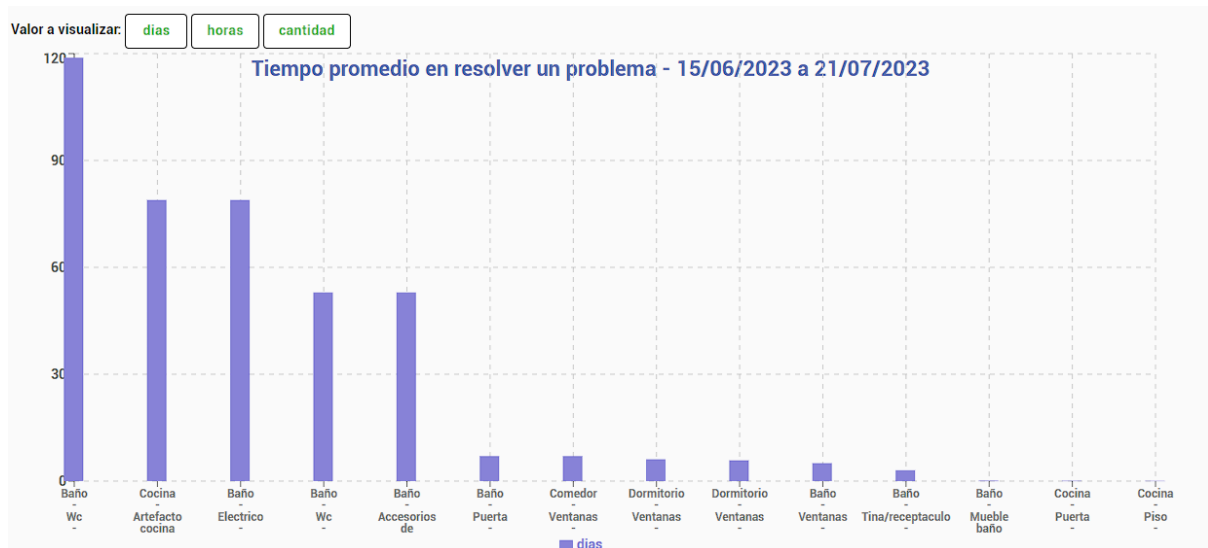


Figura 4.24: Enfoque al gráfico generado en la vista de estadísticas.

En la Figura 4.24 se hace un enfoque a la parte inferior de la vista de estadísticas, donde se muestra el gráfico generado. Se agregó un título que especifica el dato que se está visualizando, además del período seleccionado para éste. Cada gráfico cuenta con una leyenda; en el eje X se dejó el nombre del dato, mientras que en el eje Y se muestran los valores.

Por otra parte, el usuario puede seleccionar el valor a visualizar para cada dato. En el ejemplo mostrado en la Figura 4.24, se cuenta con tres posibles valores para visualizar. Esto permite tener un análisis más completo de los datos presentados, ya que en este caso a la empresa le sirve saber cuántos problemas fueron considerados para el cálculo. Por ejemplo, saber si un problema que tomó mucho tiempo resolver, solo fue reportado una vez, le permite a Magal tomar acciones diferentes que si el mismo problema fue reportado muchas veces. Por esta razón es que los datos presentados van de mayor a menor.

Como se mencionó anteriormente, al presionar uno de los botones para cambiar el tipo de gráfico, éste se actualiza automáticamente en el recuadro inferior (Fig. 4.25). A petición de Magal se dejaron 3 opciones que son consideradas las más relevantes para la visualización de estadísticas, pero ya que se trabajó con una librería bien documentada, agregar nuevos gráficos en esta vista es una tarea trivial.

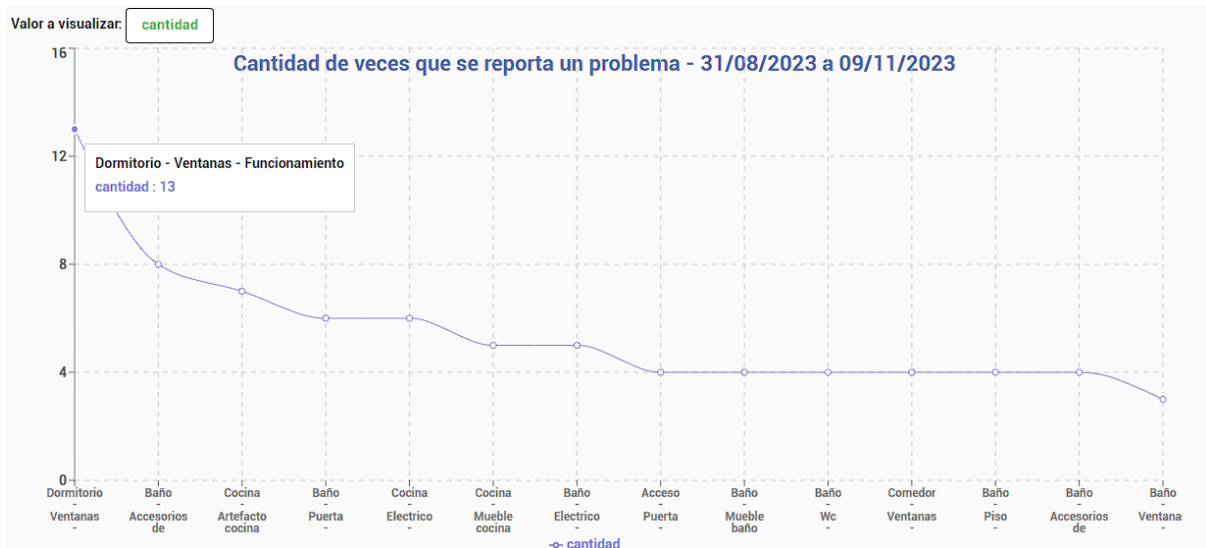


Figura 4.25: Gráfico de línea en la vista de estadísticas.

Por último, se agregó un filtro adicional (Fig. 4.26) que se muestra solo si el dato seleccionado está relacionado a problemas. Este filtro considera el tiempo promedio para resolver un problema, y la cantidad de veces que se reportan problemas. Esto permite una mayor precisión si se desea analizar tipos de propiedades o zonas específicas.

Estadística: **Cantidad de veces que se reporta un problema**

Desde: 2023-06-15 Hasta: 2023-07-21 **Actualizar**

Tipo de Propiedad: Departamento Zona afectada: Acceso Superficie afectada

Figura 4.26: Filtros adicionales en la selección de datos para estadísticas.

4.4. Paso a producción

Para integrar todos los cambios y modificaciones realizadas al sistema legado, el cual ya se encontraba en producción, se siguieron una serie de pasos. Como se definió en la memoria de Pablo Miranda [2], el sistema de post-venta/post-arriendo se mantiene en funcionamiento a través de servicios de Google Cloud Platform (GCP).

Primero, se hizo un respaldo de la base de datos actual, conectándose al sistema desde el proyecto ejecutado en local, y utilizando el comando *dumpdata* de Django para guardar toda la información en un archivo *json*. Para esto, se concedió acceso a la IP local a través del panel de comando de GCP. Luego, se creó una nueva base de datos para el sistema.

Como el sistema ya está configurado y montado para tener conexión a la base de datos del servicio GCP, solo se cambió el nombre de la base de datos previa, al de la nueva base de datos. Luego, se ejecutó el comando *migrate* de Django para actualizar los cambios a la nueva base de datos. El comando *makemigrations* se utilizó durante el desarrollo de las soluciones, por lo que el sistema local ya contaba con todas las migraciones funcionando. Con las migraciones aplicadas a la nueva base de datos, se restauraron los datos previos utilizando el comando *loaddata* y el archivo *json* anteriormente mencionado. Con esto, la nueva base de datos quedó funcionando, con los datos existentes y con las nuevas migraciones aplicadas.

Por último, para integrar los nuevos cambios en back-end y las interfaces, se instaló la aplicación Google SDK [10], que permite acceder al servicio de hosting de Google App Engine de manera rápida. Los archivos de configuración ya se encontraban presentes, y con Google SDK instalado, se utilizó el comando *gcloud app deploy* para realizar la carga de los archivos, tanto para back-end como front-end. Una vez terminado el proceso de carga, los cambios quedaron disponibles en la plataforma.

4.5. Resumen

En este capítulo se vieron tanto las decisiones en torno al diseño, como la implementación de las soluciones indicadas en el Capítulo 3. Además, se muestran las interfaces creadas e implementadas y se discuten los ajustes en torno a las propuestas del capítulo anterior.

Capítulo 5

Evaluación de la Solución

Para evaluar la usabilidad y utilidad del sistema desarrollado en este trabajo de título, se utilizó un instrumento ya conocido, el cual es la encuesta SUS (System Usability Scale) [11], la cual consta de 10 ítems que el usuario debe contestar utilizando un rango de *muy en desacuerdo* a *muy de acuerdo*, después de haber realizado las pruebas pertinentes en el sistema. Además, se evaluó la utilidad percibida de los nuevos servicios, por parte de los usuarios de Magal, empleando para ello una versión reducida de la encuesta TAM (Technology Acceptance Model) [12].

En la sección 5.1 se detallan los instrumentos de evaluación utilizados, tanto para medir la usabilidad como la utilidad percibida. A partir de los valores entregados por los trabajadores de Magal en estas encuestas, se calculan los valores de usabilidad y utilidad correspondientes. En la sección 5.2 se detalla el proceso de evaluación, y por último en la sección 5.3 se detallan los resultados obtenidos, y se discuten estos últimos a la luz de los objetivos definidos.

5.1. Detalle de los Instrumentos de Evaluación

El cuestionario que los usuarios debían responder, una vez utilizado el sistema y terminadas las tareas solicitadas, es el correspondiente a la *System Usability Scale* (SUS). En la Figura 5.1 se muestra la encuesta entregada a los usuarios. En ella, el usuario debe responder cada ítem usando una escala 1 a 5, donde 1 significa ‘muy en desacuerdo’ y 5 significa ‘muy de acuerdo’.

	1	2	3	4	5
1. Pienso que me gustaría usar frecuentemente este sistema.					
2. Encontré el sistema innecesariamente complejo					
3. Opino que el sistema fue fácil de usar					
4. Creo que necesitaría ayuda para poder usar este sistema					
5. Considero que las distintas funcionalidades estaban bien integradas					
6. Pienso que había mucha inconsistencia en este sistema					
7. Las personas aprenderían rápidamente cómo utilizar el sistema					
8. Creo que el sistema era muy complicado de usar					
9. Me sentí muy seguro al usar el sistema					
10. Necesité aprender cosas antes de usar cómodamente el sistema					

Figura 5.1: Encuesta de usabilidad de la escala SUS

A partir del cuestionario, se utiliza una fórmula para obtener un puntaje final, el cual es una referencia que permite saber si se superó el umbral mínimo para que un software se considere usable. Esta fórmula se muestra en la ecuación 1. Cabe destacar, considerando lo mostrado en la Figura 5.1, que los ítems con números impares suman más puntos si la aprobación del usuario es mayor, mientras que aquellos con números pares aumentan el puntaje si los usuarios están en desacuerdo con ellos.

$$\begin{aligned}
 \text{Items positivos} &= \text{puntaje_item} \{1, 3, 5, 7, 9\} \\
 \text{Items negativos} &= \text{puntaje_item} \{2, 4, 6, 8, 10\} \\
 \text{Puntaje Final de Usabilidad} &= 2,5 \times (\sum(\text{positivos} - 1) + \sum(5 - \text{negativos})) \quad (1)
 \end{aligned}$$

Al aplicar la fórmula anterior, se requiere que el puntaje obtenido sea mayor a 68, para que la plataforma se considere usable [13].

Para evaluar la utilidad de los nuevos servicios se les pidió a los usuarios administradores, contestar otro cuestionario. Como se mencionó antes, éste último es una versión reducida de la encuesta TAM (*Technology Acceptance Model*). En ella el usuario debe responder los ítems usando una escala de 1 a 7, donde 1 significa ‘muy en desacuerdo’ y 7 significa ‘muy de acuerdo’. Se muestra en la Figura 5.2 la encuesta entregada a estos usuarios.

	1	2	3	4	5	6	7
1. El uso de la plataforma puede mejorar el desempeño de mi labor							
2. El uso de la plataforma no facilita la realización de mi labor							
3. En general considero que la plataforma puede ser útil en mi labor							

Figura 5.2: Encuesta de utilidad entregada a los usuarios al final del estudio

5.2. Proceso de Evaluación

Los nuevos servicios desarrollados en esta memoria fueron utilizados principalmente por usuarios con rol administrador en la plataforma, mientras que solo uno de los servicios fue utilizado por usuarios de distintos perfiles. En esta sección se reportan los resultados y se muestra que los usuarios son capaces de utilizar las interfaces y realizar acciones cotidianas en la plataforma. También se puede ver que los nuevos módulos y servicios son útiles para los trabajadores de la empresa.

Para el estudio de usabilidad se trabajó con usuarios de rol “administrador”, y usuarios tipo cliente con rol de “administrador de edificios”, a fin de cubrir todos los módulos e interfaces implementadas. Se reclutaron 2 usuarios de cada rol, y a cada uno se les asignó tareas a realizar en la plataforma. A los usuarios de rol “administrador” se les asignaron más tareas, dado que hay vistas y módulos sólo accesibles para aquellos con este rol.

A grandes rasgos, a los usuarios tipo cliente se les asignaron tareas que involucran usar el módulo de registro y las interfaces disponibles a un cliente, así como también enviar tickets (reclamos a revisar o tratar). Por otra parte, a los usuarios administradores se les solicitó utilizar el módulo de carga de datos, validar clientes que se registren en el sistema y hacer uso del módulo de estadísticas.

Se evaluó a cada usuario individualmente, a los usuarios tipo cliente se les indicó asumir que era la primera vez que usaban el sistema mientras realizaban las tareas, mientras que a los usuarios administradores se les indicó considerar uso cotidiano al realizar las tareas.

Cuando los usuarios terminaban sus tareas, se les pidió contestar el cuestionario correspondiente, y entregar comentarios adicionales si lo deseaban. Esta última solo la contestó uno de los usuarios, mientras que los demás prefirieron no emitir comentarios. Por último, como se mencionó antes, los usuarios podían dejar comentarios adicionales sobre el sistema si así lo consideraban necesario.

5.3. Resultados Obtenidos

El promedio de los resultados de la evaluación de usabilidad fue de 80, siendo el puntaje más bajo 70, que es superior a 68 (valor mínimo aceptable). Los puntajes obtenidos se muestran en la Tabla 5.1. En dicha tabla se homologaron las versiones negativas de las preguntas, para mostrar puntaje equiparables (de 1 a 5) entre las diferentes columnas, para que sea más fácil de entender para el lector.

Usuario	Rol	Preguntas SUS										Puntaje SUS
		P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉	P ₁₀	
U ₁	Adm	5	2	5	4	4	4	4	4	5	3	87,5
U ₂	Adm	4	3	5	4	3	4	5	4	5	4	90
U ₃	Cliente	4	2	4	3	3	3	4	3	4	3	70
U ₄	Cliente	4	2	4	4	3	3	3	4	4	3	72,5
Suma		17	9	18	15	13	14	16	15	18	13	320

Tabla 5.1: Puntajes SUS por usuario encuestado

A partir de los resultados anteriores, se revisaron los ítems (columnas) con evaluaciones más bajas; en este caso, corresponde al ítem 2 principalmente (“encontré al sistema innecesariamente complejo”). En este caso, los usuarios creen que existe complejidad innecesaria en el sistema. Por otro lado, los ítems 5, 6 y 10 tuvieron puntuaciones levemente positivas, mientras que el resto de los ítems evaluados fueron claramente positivos en el sistema. De lo anterior se puede ver que el puntaje SUS recibido por parte de los usuarios clientes es menor al recibido por los usuarios administradores. Esto se debe a que los administradores ya están familiarizados con la estructura y el uso general de la plataforma,

mientras que los usuarios clientes al interactuar y realizar acciones en ésta por primera vez, consideran que es más compleja.

De todas maneras, los puntajes obtenidos muestran que los usuarios encuestados no consideran que el sistema fuera difícil de utilizar, o que requieran de asistencia para usarlo, pero sí creen que tiene aspectos complejos, y que puede ser simplificado. Considerando que los otros ítems negativos fueron bajos, y que los puntajes obtenidos por cada usuario son mayores a 68, se concluye que el sistema puede ser considerado “usable”.

A continuación, se analizan los resultados de la encuesta de utilidad. En la Tabla 5.2, se muestra el resultado de los puntajes obtenidos.

Usuario	Rol	Preguntas TAM			Puntaje TAM
		P ₁	P ₂	P ₃	
U ₁	Adm	7	7	7	21
U ₂	Adm	7	7	7	21
U ₃	Cliente	6	7	7	20
U ₄	Cliente	7	6	6	19
Suma		27	27	27	81

Tabla 5.2: Resultados de la encuesta TAM por usuario encuestado

En la encuesta aplicada solo un ítem estaba expresado en negativo (la pregunta 2), el cual fue expresado en positivo al igual que en el caso anterior. Los puntajes presentados están en la misma escala, y todos obtuvieron calificaciones cercanas al máximo. Eso significa que el sistema es percibido como útil por los usuarios, y que éste debería ayudar a mejorar la labor de estas personas.

Respecto a los comentarios adicionales de los evaluadores, sólo uno de ellos colocó un comentario, el cual se muestra a continuación:

El sistema puede ser mejorado, pero funciona muy bien. Con el uso se podrá plantear mejoras.

A partir de este comentario, y tomando en cuenta los resultados de ambas encuestas, se puede ver que el sistema se puede mejorar para reducir la complejidad, pero funciona correctamente y es de utilidad para los usuarios que lo utilizan.

Valoración de la Empresa

Junto a los resultados y comentarios expuestos anteriormente, que demuestran que el trabajo realizado fue percibido de forma positiva por los usuarios, se recibió un comunicado a nombre de la empresa, el cual se cita a continuación.

Los cambios hechos al sistema nos permitirán tener un registro detallado de cada atención de postventa y facilitar el seguimiento de cada uno de los casos. Esto es importante para lograr atender a cada uno de nuestros clientes en forma oportuna y eficiente.

Para nuestra empresa el servicio es muy importante, estos cambios y mejoras ayudarán a que este atributo sea destacado cada vez que un cliente requiera nuestra atención.

El trabajo de Sebastián ha sido impecable y profesional, entendiendo que no solo es importante el desarrollo y funcionalidades del sistema, sino que también el cómo lo perciben nuestros clientes, y qué tan amigable es para nuestros colaboradores, ya que ellos serán los usuarios en el día a día.

Además, esta plataforma será la cara visible de nuestra empresa con todos nuestros clientes, es importante que demuestre nuestro profesionalismo y calidad de servicio como Grupo Magal.

Con lo presentado en este capítulo, es posible decir que se da por cumplido el objetivo específico 4 de este trabajo de memoria, presentado en la sección 1.3. Es decir, la plataforma cumple con la usabilidad y utilidad esperadas, permitiendo un uso aceptable para Magal.

Capítulo 6

Conclusiones y Trabajo a Futuro

El trabajo realizado en esta memoria permitió a la empresa, por primera vez, contar con un registro de usuarios para los clientes de su inmobiliaria. Además, permitió un mayor manejo de la información en su base de datos, y la visualización de estadísticas con gráficos generados a partir de todos los reportes y registros de visitas. Con estos servicios se espera poder tener una mayor claridad al momento de tomar decisiones.

De acuerdo con lo expuesto en esta memoria, se cumplieron todos los objetivos específicos. Se hizo un análisis de los problemas presentes en el sistema legado, y a partir de éste se diseñó una solución que resuelve las deficiencias del sistema anterior.

Respecto a la validación del nuevo sistema, los participantes quedaron satisfechos con los resultados del trabajo, y se concluyó que las interfaces y los servicios cumplen con la usabilidad esperada. Además, el sistema en su conjunto cumple con la utilidad esperada para que los usuarios puedan realizar sus labores.

El objetivo general de la memoria fue cumplido, pues el nuevo sistema permite a Magal realizar la carga de nuevos edificios, sin tener que recurrir a un usuario externo. Además, los clientes de la empresa ahora cuentan con un registro que les permitirá crear tickets con mayor seguridad, así como también tener un mejor control sobre aquellos tickets que han reportado. Por último, la empresa cuenta ahora con un panel de estadísticas donde podrán ver las actividades recientes y pasadas, para así tener un mejor análisis de los datos en el sistema.

Respecto al trabajo a futuro, se puede destacar la necesidad de modificar el módulo de carga de datos, para reducir la cantidad de recuadros con los que el usuario puede interactuar. Para esto, el sistema debería permitir al usuario seleccionar cuáles hojas del archivo subido se quiere procesar, y así realizar la carga considerando esa selección. Esto eliminaría la necesidad de contar con recuadros para procesar hojas de forma individual.

Otro trabajo a futuro consiste en permitirle a la empresa rechazar los problemas de forma individual, de entre aquellos reportados en un ticket. Si bien se hizo un cambio en el modelo de datos para permitir esta funcionalidad, la misma no se implementó durante el desarrollo de la memoria debido a razones de tiempo y a que dicha funcionalidad era opcional. Por lo tanto, a futuro el sistema debería permitir aprobar o no los problemas al momento de revisar un ticket pre-aprobado, e incluir una explicación que indique al cliente el motivo por el que se rechazó el o los problemas.

Por otra parte, en cuanto a seguridad respecta, se puede agregar una mayor trazabilidad sobre las acciones de los usuarios en el sistema. Actualmente se cuenta con registro sobre cuáles usuarios realizan ciertas acciones, por ejemplo, al cerrar un ticket, al realizar una visita, entre otros, pero estas se almacenan en las tablas respectivas. Se puede crear una nueva tabla que guarde un usuario y su acción realizada, a fin de poseer un registro y seguimiento más completo sobre lo que realizan los usuarios en el sistema.

Otro aspecto a mejorar es que el sistema revise de forma automática si el plazo del servicio de post-venta para cada propiedad venció o no. El sistema ya cuenta con tareas que se ejecutan de forma diaria, por lo que se puede programar una nueva tarea que revise y cambie el estado de una propiedad si el plazo ya venció.

Por otro lado, se puede mejorar el diseño del panel de estadísticas para acomodar los recuadros, como así incluir más opciones de gráficos. A su vez, se pueden agregar nuevos datos a visualizar que servirían a Magal para realizar otros tipos de análisis.

Un aspecto a considerar, es que la empresa no dispone de ambientes separados para el software de post venta. Solo existe un ambiente de desarrollo, por lo que la empresa carece de ambientes de testing y deployment. Esto presenta limitaciones y riesgos, por lo que una práctica recomendable para la empresa sería crear ambientes para disminuir los riesgos al momento.

Otro aspecto es modificar las vistas para que sean responsivas. Si bien la mayoría de las vistas serán utilizadas en navegador a través de un PC de escritorio, sería útil para los trabajadores de Magal que estas se puedan visualizar en otros dispositivos, ya sean teléfonos o tablets.

Por último, un aspecto importante que sería útil para la empresa, sería contar con una interfaz que permita exportar datos del sistema y descargarlos en un formato accesible y fácil de leer, por ejemplo en formato Excel. Esta interfaz debería permitir al usuario seleccionar cuáles datos quiere exportar y si los desea con alguna estructura u orden específicas. La información descargada sería de gran utilidad para Magal.

Bibliografía

- [1] M. Navas. ¿Será 2023 el año del repunte inmobiliario? Expertos proyectan qué se espera en el sector. 2022. URL: <https://dfmas.df.cl/df-mas/como-cuido-mis-lucas/sera-2023-el-ano-del-repunte-inmobiliario>. Acceso: 2023-12-15.
- [2] P. Miranda Álvarez. Sistema de alerta preventiva y CRM para Inmobiliaria MAGAL. Memoria de ingeniería Civil en Computación. 2020. DCC, Universidad de Chile. URL: <https://repositorio.uchile.cl/handle/2250/176629>. Acceso: 2023-12-15.
- [3] J. Walke. React, a free and open-source front-end JavaScript library. 2013. <https://legacy.reactjs.org/>. Acceso: 2023-12-15.
- [4] A. Clark. Pillow (PIL Fork) Documentation. Readthedocs. 2015. Retrieved from <https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf>. Acceso: 2023-12-15.
- [5] W. McKinney, et. al. Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference, Vol. 445, pp. 51–56. 2010.
- [6] M. Henk, R. Sass, T. Rein, U. Purde, M. Tajur. Pipedrive | every deal matters. <https://www.pipedrive.com/es>, 2010. Acceso: 2023-12-17.
- [7] Roy Mann and Eran Zinman. monday.com | every deal matters. <https://monday.com/lang/es>, 2012. Acceso: 2023-12-17.
- [8] BReal Team. Operación y mantenimiento de edificios. <https://www.breal.cl/administracion-de-mantenimiento>, 2014. Acceso: 2023-12-17.
- [9] H. Emekoma. The top 8 react chart libraries. 2023. <https://blog.logrocket.com/top-8-react-chart-libraries/>. Acceso: 2023-12-15.
- [10] Google Inc. Cloud SDK documentation. <https://cloud.google.com/sdk/gcloud/reference>, 2020. Acceso: 2023-12-17.
- [11] J. Brooke. SUS: A quick and dirty usability scale. Usability Eval. Ind. 189. 1995. Acceso: 2023-12-15.
- [12] F. D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly 13 (3): 319-340, 1989. doi:10.2307/249008. Acceso: 2023-12-15.
- [13] Jeff Sauro. Measuring usability with the system usability scale (sus). <https://measuringu.com/sus/>, 2011. Acceso: 2023-12-15.

- [14] Adrian Holovaty and Jacob Kaplan-Moss. The Definitive Guide to Django: Web Development Done Right, Second Edition. Apress, Berkely, CA, USA, 2nd edition, 2009.
- [15] Grinberg, M., 2018. Flask web development: developing web applications with python. [https://en.wikipedia.org/wiki/Flask_\(web_framework\)](https://en.wikipedia.org/wiki/Flask_(web_framework)). Acceso: 2023-12-15
- [16] Mindfire Solutions. Flask vs django. <http://www.mindfiresolutions.com/blog/2018/05/flask-vs-django/>, 2018. Acceso: 2023-12-15.