



**Aplicación web para la búsqueda centralizada de casas y apartamentos
en las principales ciudades del país.**

House Finder

Luisa Fernanda Aldana Gallo

10892314888

Edwin Alexander Bohórquez Gamba

10892316049

Universidad Antonio Nariño

Programa Especialización en Ingeniería de Software

Facultad de Sistemas

Bogotá, Colombia

2023

**Aplicación web para la búsqueda centralizada de casas y apartamentos
en las principales ciudades del país.**

House Finder

Luisa Fernanda Aldana Gallo

Edwin Alexander Bohórquez Gamba

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

Director (a):

Dianalin Neme Prada

Universidad Antonio Nariño

Programa Especialización en Ingeniería de Software

Facultad de Sistemas

Bogotá, Colombia

2023

NOTA DE ACEPTACIÓN

El trabajo de grado titulado

_____, Cumple con

los requisitos para optar

Al título de _____.

Firma del Tutor

Firma Jurado

Firma Jurado

Bogotá, 09 de noviembre de 2023.

Tabla de contenido

Contenido

Tabla de contenido	2
Tabla de ilustraciones	5
Tabla de tablas.....	6
Resumen	7
Abstract	8
1. Introducción	9
2. Formulación y descripción del problema	11
3. Objetivos	15
3.1. Objetivo General.....	15
3.2. Objetivos específicos:.....	15
4. Marco de referencia	16
4.1. Estado del arte.....	16
4.2. Impacto	19
4.3. Componente de innovación	19
5. Marco Teórico	21

5.1. API RESTful:	21
5.2. Uso de datos abiertos en Colombia	22
5.3. ¿Por qué hablamos de datos abiertos?	23
5.4. Ley de habeas data	23
5.5. Herramientas	24
5.6. Trello	24
5.7. Tecnologías	25
5.8. Arquitectura del proyecto	27
5.9. Metodología	28
6. Metodología	30
7. Proceso de software	32
7.1. Análisis de requerimientos	32
7.1.1. Requerimientos funcionales	32
7.1.2. Requerimientos no funcionales	33
8. Diseño	34
8.1. Creación del prototipo	34

9. Diseño y arquitectura	41
10. Construcción.....	54
11. Instalación y configuración	56
12. Pruebas y calidad de código	60
13. Conclusiones	63
14. Referencias.....	64

Tabla de ilustraciones

Ilustración 1	10
Ilustración 2	12
Ilustración 3	14
Ilustración 4	20
Ilustración 5	31
Ilustración 6	35
Ilustración 7	36
Ilustración 8	38
Ilustración 9	39
Ilustración 10	40
Ilustración 11	41
Ilustración 12	42
Ilustración 13	43
Ilustración 14	43
Ilustración 15	44
Ilustración 16	45
Ilustración 17	46
Ilustración 18	47
Ilustración 19	48
Ilustración 20	49

Ilustración 21	49
Ilustración 22	51
Ilustración 23	52
Ilustración 24	53
Ilustración 25	56
Ilustración 26	57
Ilustración 27	58
Ilustración 28	59
Ilustración 29	60
Ilustración 30	61
Ilustración 31	62

Tabla de tablas

Tabla 1	13
Tabla 2	16
Tabla 3	17
Tabla 4	32
Tabla 5	33
Tabla 6	55

Resumen

El presente documento presenta el proyecto “House Finder” para la especialización en ingeniería de software de la Universidad Antonio Nariño que tiene por objetivo centralizar la búsqueda de inmuebles en las principales ciudades de Colombia a través de un aplicativo web, diseñado para personas interesadas en compra venta o arriendo de inmuebles.

El proyecto se desarrolló bajo la metodología Kanban, la cual permite conocer el estado general del trabajo que se tiene que realizar, así como la prioridad de las tareas. Se implementó en lenguaje de programación Java para el desarrollo de backend y para todo lo referente con el frontend se desarrolló en el framework Angular.

Abstract

The present document details the "House Finder" project for the specialization in software engineering at the Universidad Antonio Nariño, which aims to centralize the search for real estate in the main cities of Colombia through a web application, designed for people interested in buying, selling, or renting real estate.

The project was developed using the Kanban methodology, which allows for tracking the overall status of work to be done, as well as the priority of tasks. It was implemented in the Java programming language for backend development, and the frontend was developed using the Angular framework.

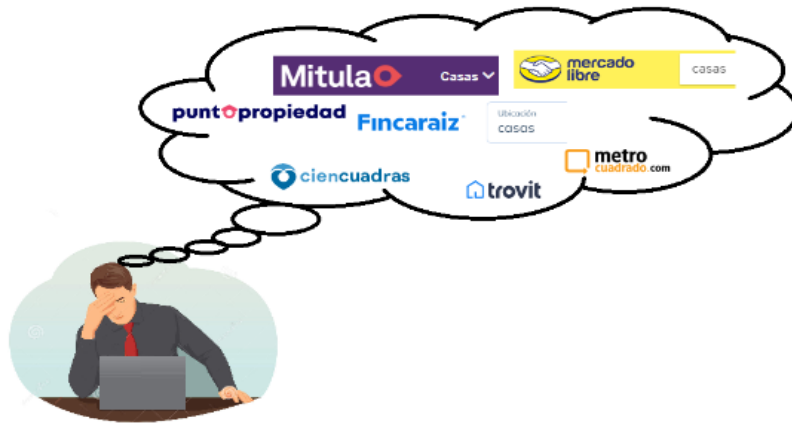
1. Introducción

Cuando una persona va a comprar una casa se ve en la necesidad de ir a diferentes inmobiliarias y recibir asesoría de agentes para la compra del inmueble o como segunda opción el usuario tiene las plataformas virtuales de finca raíz que disponen de mucha información para que esta pueda ser consultada. Cualquiera de estas opciones se puede convertir en un proceso agotador para el usuario ya que él debe invertir mucho tiempo en buscar fuentes confiables y actualizadas, filtrar por diferentes parámetros, clasificar mucha información, guardar ofertas en las diferentes aplicaciones, llevar un registro manual, realizar una comparación entre las opciones guardadas, visitar diferentes casas y muchas veces ver ofertas que al final no cumplen con sus expectativas.

Por este motivo se diseñó House Finder, un aplicativo web que permitirá a los usuarios visualizar las mejores ofertas inmobiliarias en un solo lugar. Su funcionamiento se basa en unos parámetros de búsqueda que la aplicación recibirá y se encargará de validarlos en 3 de las principales páginas de venta de casas, dando como resultado las mejores opciones de inversión inmobiliaria. Con ello el usuario ahorrará tiempo en su búsqueda y tendrá toda la información en un solo lugar para poderla comparar fácilmente.

Ilustración 1

Usuario pensando en buscar casa



Fuente: Elaboración Propia

Nota: La imagen representa a un usuario pensando en buscar una casa en las principales plataformas inmobiliarias.

2. Formulación y descripción del problema

¿Cómo optimizar la búsqueda de inmuebles con el uso de aplicaciones web de venta de finca raíz para obtener las mejores ofertas en corto tiempo y en una sola respuesta?

Si alguna vez ha buscado un apartamento o comprado una casa a través de herramientas digitales sabe lo mucho que hay que clasificar para lograr dicho objetivo. Actualmente hay muchas plataformas que permiten al usuario buscar inmuebles, pero debe ingresar a cada una de ellas los parámetros de búsqueda y filtrar entre los resultados los que más se aproximen al inmueble deseado. Todo este proceso implica invertir mucho tiempo y suele ser bastante desgastante, adicional que ninguna plataforma ofrece la opción de centralizar la búsqueda. Este proyecto selecciona las tres herramientas más usadas según información brindada por la página: similarweb.com, las cuales históricamente han sido las más importantes para la búsqueda de finca raíz en Colombia.

Ilustración 2

Ranking de páginas usadas en Colombia para búsqueda de inmuebles

1	 metrocuadrado.c...	Empresas y servicios al consumidor > Inmobiliaria
2	 fincaraiz.com.co	Empresas y servicios al consumidor > Inmobiliaria
3	 ciencuadras.com	Empresas y servicios al consumidor > Inmobiliaria
4	 mitula.com.co	Empresas y servicios al consumidor > Inmobiliaria
5	 supapush.net	Empresas y servicios al consumidor > Inmobiliaria

Fuente: Similarweb (2023)

Nota: La imagen informa las plataformas más usadas en el mes de septiembre de 2023 para la búsqueda de inmuebles en Colombia.

Como estudio investigativo se validó en los tres primeros portales del ranking de portales inmobiliarios más importantes de Colombia, cuanto es el tiempo que toma realizar una búsqueda, aplicar filtros de acuerdo a las preferencias del usuario, guardar y comparar inmuebles. Lo anterior permitió validar que en todos los portales no existe una forma de filtrar desde el inicio las preferencias de búsqueda, sino que deben ser modificadas una vez se realiza la búsqueda en los portales. Adicionalmente, el usuario debe revisar cada opción para validar si se

acomoda a sus necesidades lo que aumenta los tiempos en la búsqueda, a continuación se relaciona la tabla con los tiempos tomados.

Tabla 1

Tabla investigación tiempos de búsqueda de casas en portales.

Aplicativo	Respuesta Búsqueda (segundos)	Filtrar criterios (segundos)	Analizar Resultados (minutos)	Total (minutos)
Finca Raíz	0,05	5	35	40,05
Metrocuadrado	0,05	6	30	36,05
Cien Cuadras	0,07	10	45	55,07

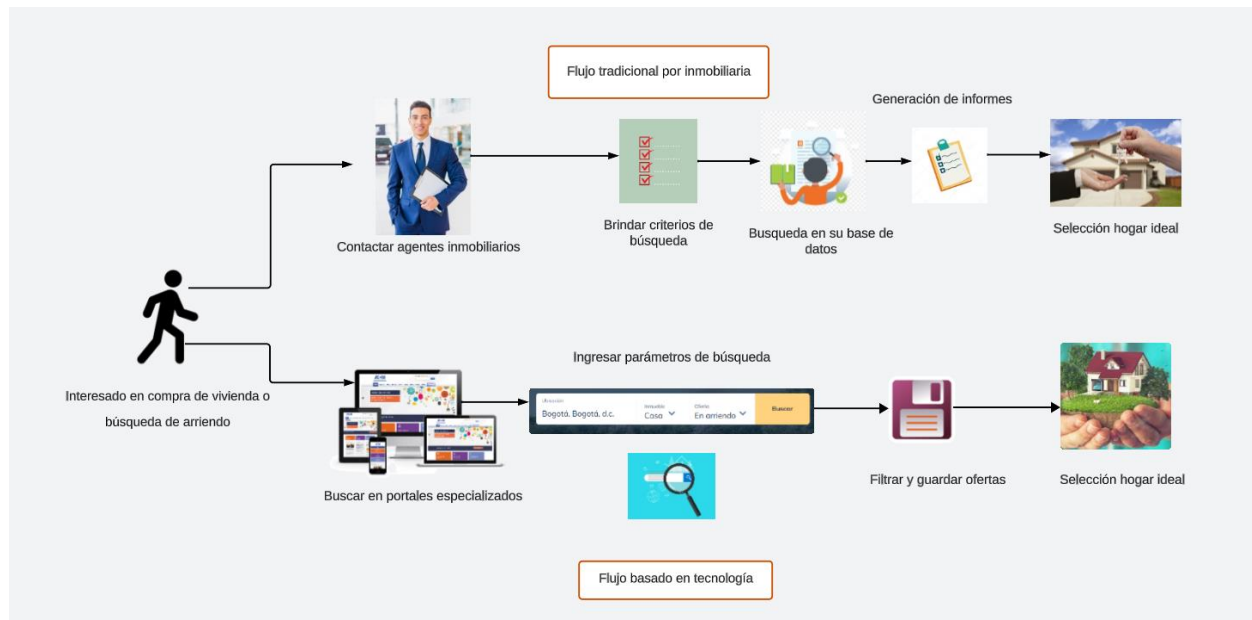
Fuente: Elaboración Propia

Nota: La tabla brinda el resultado de los tiempos invertidos en la búsqueda de inmuebles en los 3 portales más importantes.

Flujo del proceso:

Ilustración 3

Descripción del flujo actual que sigue un usuario para compra de



Fuente: Elaboración propia

Nota: El gráfico representa las dos alternativas que tiene un usuario para buscar casa. En cada una de ellas se muestran todas las fases por las que pasa el usuario.

3. Objetivos

3.1. Objetivo General

Crear una aplicación web que realice búsquedas automáticas en las plataformas: finca raíz, metro cuadrado y cien cuabras y brinde al usuario las ofertas que más se ajusten a sus necesidades.

3.2. Objetivos específicos:

- Realizar una búsqueda automatizada con la información del inmueble dada por el usuario, en Finca raíz, Metro cuadrado y cien cuabras.
- Comparar los resultados obtenidos de las plataformas inmobiliarias ya mencionadas y brindar las ofertas que más se ajusten a las necesidades del usuario.
- Generar un listado de ofertas inmobiliarias que se ajusten a los parámetros ingresados por el usuario mostrando el resultado de menor a mayor precio.
- Reducir el tiempo de búsqueda de un inmueble.

4. Marco de referencia

4.1. Estado del arte

En la actualidad existen infinidad de páginas dedicadas a la publicación y difusión de inmuebles disponibles para compra y venta, algunas permiten realizar comparaciones entre los mismos, otras ofrecen búsquedas personalizadas y brindan consejos para mantener informado al usuario. Al final todas ellas buscan por medio de la tecnología brindar un contacto directo con el vendedor, sea parte de una inmobiliaria o sea un vendedor independiente. A continuación, se relacionan algunos de los principales portales de vivienda usados en Colombia:

Tabla 2

Tabla descriptiva de plataformas de vivienda existentes en Colombia

Nombre	País origen	Descripción
Metrocuadrado	Colombia	Portal enfocado en la oferta de venta y arriendo de inmuebles en Colombia, funciona como clasificados.
Finca Raíz	Colombia	Portal enfocado en la búsqueda y publicación de inmuebles en Colombia, funciona como clasificados.
Ciencuadras	Colombia	Portal enfocado en la búsqueda de inmuebles en venta o arriendo en toda Colombia, es aliada de aseguradora Bolívar.
Estrenar Vivienda	Colombia	Portal que presenta la oferta de proyectos de vivienda nueva en Bogotá y en las principales ciudades de Colombia.
MiTula	España	Es un buscador de clasificados enfocado en varios ámbitos como inmuebles, vehículos y empleo, con presencia en 49 países.

Fuente: Elaboración propia

Nota: La tabla presenta descripción de varios portales web destinados a publicaciones inmobiliarias.

Adicionalmente existen diversos portales enfocados en el comercio electrónico que sirven como intermediarios para que se publiquen y visualicen estos inmuebles, es el caso Mercado Libre o marketplace, los cuales son muy usados por los colombianos para esta tediosa tarea de búsqueda de inmuebles.

A continuación, se relacionan diferentes portales enfocados en búsquedas de servicios particulares que nos permiten tomar un marco de referencia para nuestro proyecto:

Tabla 3

Tabla descriptiva de plataformas buscadoras de servicios.

Nombre	Sector	País de origen	Descripción
Trivago	Turístico	Alemania	Plataforma enfocada en la búsqueda y comparación de servicios de alojamiento, se encuentra categorizado como un metabuscador.
Booking	Hotelero	Ámsterdam	Plataforma enfocada en la búsqueda y comparación de servicios de alojamiento.
Skyscanner	Vuelos / transporte	Reino Unido	Plataforma enfocada en la búsqueda y comparación de vuelos y transporte.

Nombre	Sector	País de origen	Descripción
Indeed	Empleo	Estados Unidos	Plataforma enfocada en la búsqueda de empleo a nivel mundial
PinBus.com	Transporte terrestre	Colombia	Plataforma para buscar, comparar y comprar tiquetes de transporte terrestre en Colombia.

Fuente: Elaboración propia

Nota: La tabla presenta descripción de varios portales web destinados a búsqueda y comparación de servicios.

Después de verificar las plataformas mencionadas en la tabla 2 y otras adicionales, se identificó que en todos los portales no existe una forma de filtrar desde el inicio las preferencias de una búsqueda, adicionalmente se debe ingresar en cada opción brindada en la respuesta para validar si se ajusta a las necesidades, lo cual aumenta los tiempos en la búsqueda. También se identificó que no existe un portal que centralice las búsquedas para búsqueda de inmuebles, por ese motivo se diseña House Finder, para suplir esta necesidad.

4.2. Impacto

4.2.1. Búsqueda automatizada. Con House Finder el usuario ya no tendrá que buscar en cada plataforma de venta de inmuebles, sino que en un solo lugar podrá obtener resultados de las páginas inmobiliarias más importantes del país.

4.2.2. Reducción de tiempos en la búsqueda de casas. Teniendo en cuenta que la búsqueda será automatizada, el usuario ya no tendrá que invertir tiempo en: buscar en cada plataforma, guardar un registro de cada inmueble, comparar precios, sino que tendrá las ofertas que más se ajusten a sus necesidades en un tiempo adecuado.

4.2.3. Selección de las mejores ofertas para el usuario. Por medio de agentes de búsqueda, House Finder se encargará de visitar las plataformas inmobiliarias más importantes del país y extraer las ofertas que más se ajusten a las necesidades del usuario.

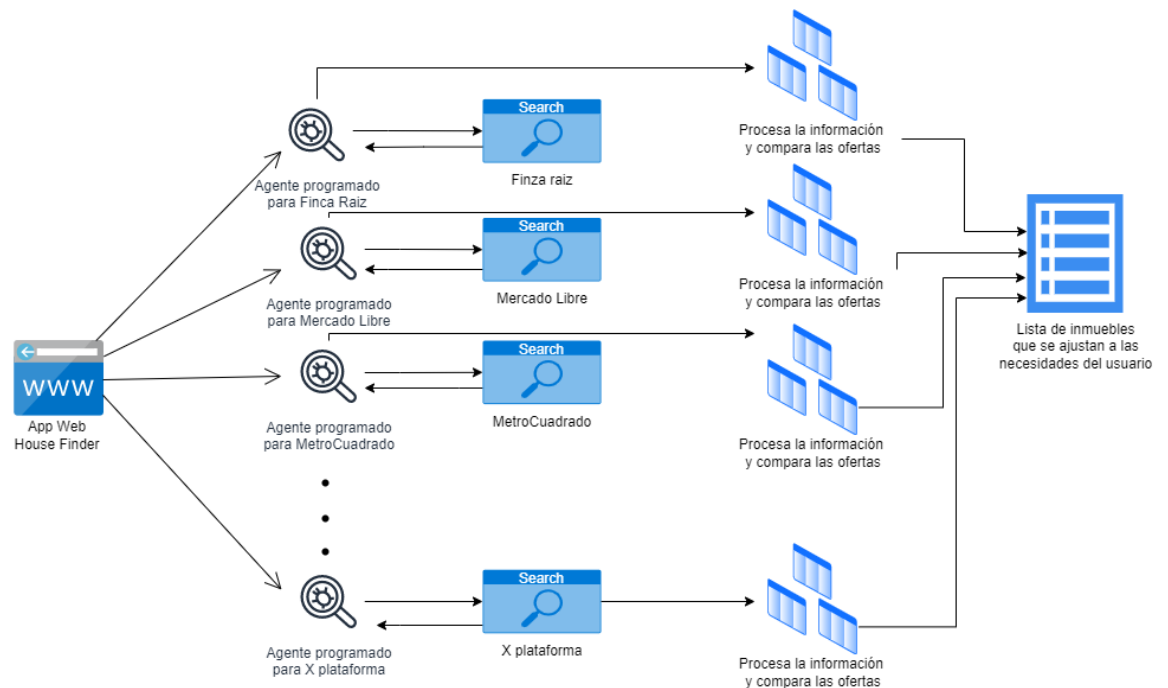
4.3. Componente de innovación

Por medio de las Apis que expone metro cuadrado, finca raíz y ciencuadras, se extraerá información y se pondrá a disposición del usuario para que

pueda ver y comparar las ofertas que más se ajusten a sus necesidades. Adicionalmente, pensando en el tiempo del usuario, este proceso hará una búsqueda paralela en cada plataforma (por medio de agentes de software programados) para que al final del proceso se tenga una respuesta en un tiempo adecuado.

Ilustración 4

Búsqueda en paralelo en las diferentes plataformas inmobiliarias



Fuente: Elaboración propia

Nota: El gráfico muestra a nivel general todas las acciones que se realizarán en House Finder para obtener las ofertas inmobiliarias que más se ajusten a las necesidades del usuario.

5. Marco Teórico

5.1. API RESTful:

Para lograr el objetivo principal de nuestra propuesta, se utilizará la conexión a través de tres interfaces API RESTful, a continuación, se detalla su funcionalidad, importancia y estado legal en Colombia.

¿Qué es API Restful?

Una API es una interfaz de programa de aplicación en la cual permite exponer datos y funcionalidades entre programas para facilitar sus interacciones, en esta se fijan qué reglas se deben usar para efectuar una comunicación con otros sistemas de software. Por otro lado, Rest es una arquitectura de software que se aplica al diseño de API y especifica qué condiciones deben seguir para su correcto funcionamiento. Es decir que al usar el término REST API hablamos de una interfaz que utiliza un protocolo seguro que permite la conexión entre varios sistemas para compartir información a través de internet, usa formatos como XML, JSON, entre otros para generar la conexión. (*Masse, 2011*)

(2)

En la actualidad existen muchas páginas que nos brindan servicios web para el desarrollo de aplicaciones de uso libre, esto nos permite experimentar con datos reales a través del consumo de API de manera sencilla y así ejecutar pruebas cuando se está desarrollando una nueva aplicación. Por otro lado, los portales empresariales suelen usar las API como generadores de valor y crecimiento ofreciendo servicios digitales a terceros, en los cuales se pueden consultar de manera sencilla, esto permite generar comunidades de desarrolladores, facilitar el acceso a nuevos clientes, crear nuevas fuentes de ingresos entre otros. (Benzell et al., 2019) (3)

5.2. Uso de datos abiertos en Colombia

“Los datos abiertos son datos que cuentan con las características técnicas y legales necesarias que pueden ser utilizados, reutilizados y redistribuidos libremente por cualquier persona en cualquier momento y en cualquier lugar.” Osorio-Sanabria et al. (2021) (4)

“La apertura de datos públicos es el proceso que pone al alcance de la sociedad los datos, en formas digitales y abiertas, esto permite aumentar el valor social, político y económico.”

Garriga-Portolà, M. (2011). (5)

5.3. ¿Por qué hablamos de datos abiertos?

El gobierno colombiano ha dispuesto a la comunidad el portal www.datos.gov.co donde encontramos una amplia base de datos en diferentes temas, para ser usados en investigación, desarrollo y visualizaciones. Actualmente, en Colombia, la apertura de datos públicos está regulada por la Ley 1712 de 2014, sobre transparencia y derecho de acceso a la información pública, que tiene como objetivo *“regular el derecho de acceso a la información pública, los procedimientos para el ejercicio y garantía del derecho y las excepciones a la publicidad de información”*. *González Tocarruncho, J. F. (2022). (6)*

5.4. Ley de habeas data

En 2012, se aprobó la ley 1581, también conocida como la ley de protección de datos personales. Su propósito fundamental es materializar el derecho constitucional que poseen todas las personas a estar informadas, actualizar y corregir la información recopilada sobre ellas en bases de datos o archivos. *“Esta ley se generó con el fin de salvaguardar la información personal o delicada de individuos naturales y jurídicos.” Cifuentes Muñoz, E. (1997). (7)*

Se expone esta ley en este documento ya que los tres portales inmobiliarios usados en nuestro proyecto (metro cuadrado, finca raíz y ciencuadras) tienen definidas

políticas informando a la comunidad el alcance y tratamiento de datos sensibles recolectados en sus publicaciones, por este motivo House Finder también se guía y rige dentro del marco legal colombiano indicado en el punto 2.4.

5.5. Herramientas

5.6. Trello

“Trello es una aplicación visual diseñada para facilitar la gestión de diversos proyectos y flujos de trabajo, así como el seguimiento de tareas por parte de los equipos. Brinda la posibilidad de incorporar archivos, listas de verificación e incluso automatizaciones, permitiendo una personalización completa de acuerdo con las necesidades específicas del equipo”. (Trello, 2023).

(8)

Trello dispone de tableros, listas y tarjetas para que el equipo pueda definir asignaciones, ver quien está trabajando y saber que pendientes o cuellos de botella presenta el proyecto.

Elementos básicos de Trello:

Tableros: Dentro de un tablero se pueden crear diferentes listas o fases que ayudarán a mantener las tareas organizadas y a que el trabajo avance. En un tablero se podrá ver todo el estado del proyecto, desde las tareas que hay por hacer hasta las asignaciones que fueron completadas.

Listas: Definen las fases por las que puede pasar una tarea. Según el flujo de trabajo o de acuerdo a las necesidades de negocio se puede crear varias listas que representen las etapas por las cuales debe pasar cada asignación.

Tarjetas: Son requerimientos, tareas o ideas que tienen la información necesaria para hacer el trabajo. Cada vez que haya un avance, las tarjetas deben moverse de una lista a otra para indicar el estado de trabajo en el cual se encuentran.

5.7. Tecnologías

Teniendo en cuenta que se aplicaran técnicas para la extracción de información, se eligió el lenguaje de programación Java ya que este posee librerías

y utilidades que soportan la búsqueda y el análisis de información que proviene de la web.

- **Java:** Es un lenguaje de programación y plataforma informática que tuvo su origen en 1995 de la mano de Sun Microsystems. *“Una de las características más destacadas de Java radica en su habilidad para trasladarse de manera sencilla entre diversos sistemas informáticos. La capacidad de ejecutar un mismo programa en múltiples sistemas resulta fundamental para el software destinado a la World Wide Web”.* (IBM, 2023). (9)

Java es un lenguaje de programación orientado a objetos, con el cual se podrá modelar la estructura de datos de: casas, apartamentos, ofertas y agentes de búsqueda que se encargaran de extraer la información de las plataformas inmobiliarias.

- **Rest:** Es un estilo de arquitectura de software que se emplea para vincular diversos sistemas a través del protocolo HTTP. El aplicativo intercambiará información con sistemas externos por medio de servicios Rest y los datos se enviarán en formato JSON.

- Angular: El diseño del frontend se realizará con el framework Angular, este será un portal de tipo progressive web app (PWA) que se ajustará a versiones móviles y web a través de componentes visuales de Angular material. La interacción desde el frontend hacia el backend se ejecutará a través de TypeScript integrado también en el framework por medio de archivos JSON.

5.8. Arquitectura del proyecto

En el proyecto se implementará la arquitectura de Microkernel, también conocida como arquitectura de Plug-in. En dicho estilo arquitectónico las aplicaciones se dividen en dos tipos principales de componentes: el sistema Core (o núcleo del sistema) y los plugins (o módulos). *“El sistema Core contiene los elementos esenciales para que la aplicación funcione y cumpla con su propósito original. Por otro lado, los módulos o plugins son componentes adicionales que se incorporan o instalan en el componente Core para ampliar su funcionalidad. En este contexto, se permite la existencia de un único componente Core, pero pueden haber múltiples plugins”.* (Oscar Blancarte, 2023). (10)

Teniendo en cuenta que el proceso de búsqueda en una plataforma inmobiliaria se puede ver como una funcionalidad independiente, éste se puede

considerar como un plugin o un módulo que se puede integrar a la aplicación. Es por ello que se seleccionó esta arquitectura porque permitirá que la aplicación sea extensible ya que si se quisiera agregar más adelante una nueva funcionalidad (buscar en una nueva plataforma inmobiliaria), el core del aplicativo no se afectaría bruscamente, pues dicha funcionalidad se agregaría como un plugin que extienda la funcionalidad inicial del sistema.

5.9. Metodología

Para gestionar el proyecto se implementará la metodología Kanban ya que es una metodología ágil y flexible que permitirá visualizar el flujo de trabajo e identificar posibles cuellos de botella.

Por definición: *“Kanban es un sistema visual para gestionar el trabajo a medida que avanza en un proceso. Kanban visualiza tanto el proceso (el flujo de trabajo) como el trabajo real que pasa por ese proceso. El objetivo de Kanban es identificar posibles cuellos de botella en el proceso y solucionarlos para que el trabajo pueda fluir a través de él de forma rentable y a una velocidad o rendimiento óptimos”*. (Kanban, 2023). (11)

Esta metodología se caracteriza por principios fundamentales y seis prácticas básicas, las cuales son:

Principios fundamentales:

1. Aceptar cambios de manera gradual y evolutiva
2. Respetar las funciones, responsabilidades y roles existentes en un principio
3. Promover acciones de liderazgo en todos los niveles

Prácticas básicas:

- Visualizar el proceso de trabajo
- Restringir el trabajo en curso (WIP)
- Administrar el flujo
- Explicitar las políticas de los procedimientos
- Incorporar bucles de retroalimentación
- Mejorar a través de la colaboración y la evolución experimental

6. Metodología

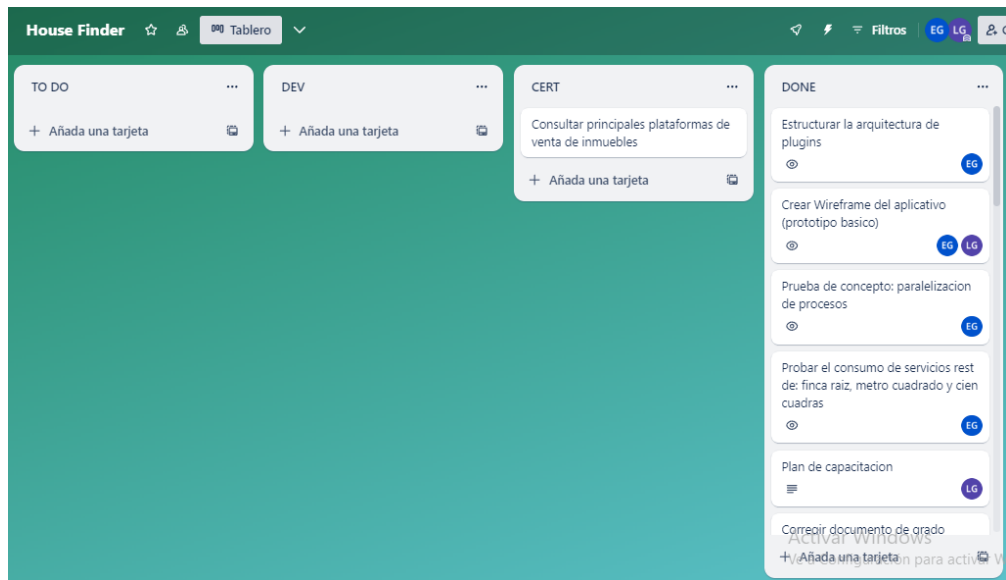
Tomando como base la metodología de Kanban, se manejan iteraciones de 2 semanas en las que cada miembro del equipo adoptará las funciones siguientes:

- Product Manager. Encargado de gestionar los requisitos dentro del sistema.
- Delivery Manager. Encargado del flujo de trabajo y de facilitar las reuniones de planeación y revisión.
- Developer. Encargado de implementar y cumplir con el requerimiento que se incluya en cada tarjeta del tablero Kanban.

Se manejará el flujo de trabajo bajo el siguiente tablero:

Ilustración 5

Tablero Kanban



Fuente: Elaboración propia

Nota: Tablero Kanban creado en la herramienta Trello

En dicho tablero se pueden observar 5 fases las cuales almacenarán cada unidad de trabajo por medio de tarjetas y cada lista tendrá un límite de 5 tareas a excepción de la lista TO DO que tendrá todos los requerimientos del sistema. Con esto se prevendrán posibles obstrucciones en el proceso de trabajo. Una vez se pase una tarjeta a DONE, esta será evaluada por el equipo y se hará una revisión para verificar que haya cumplido con los criterios de aceptación.

7. Proceso de software

7.1. Análisis de requerimientos

7.1.1. Requerimientos funcionales

Tabla 4

Tabla informativa de los requerimientos funcionales del sistema.

ID	Nombre del requerimiento	Descripción	Parámetros
RF1	Pedir los datos de la casa o apartamento que busca el usuario.	El sistema debe permitir ingresar los parámetros de búsqueda	Ciudad Tipo de inmueble (casa o apartamento) Presupuesto (rango de valores (valor mínimo y valor máximo)) Número de habitaciones Número de baños Número de parqueaderos Área (rango de valores (valor mínimo y valor máximo))
RF2	Consultar principales plataformas de venta de inmuebles	El sistema debe buscar simultáneamente en los siguientes portales	Finca raíz Metro Cuadrado Cien Cuadras
RF3	Comparar los resultados obtenidos de la búsqueda teniendo como prioridad el precio y las características solicitadas por el usuario	El sistema debe comparar los inmuebles de las 3 plataformas y teniendo en cuenta los parámetros ingresados por el usuario, retornar los inmuebles que más se ajusten al precio del usuario.	
RF4:	Listar las casas o apartamentos que se ajustan a las necesidades del usuario.	El sistema debe listar todos los inmuebles que cumplan con los parámetros ingresados por el usuario, en caso de no encontrar inmuebles se debe mostrar que no se encontraron resultados.	

Fuente: Elaboración propia

Nota: La tabla presenta la información de los requerimientos funcionales que debe cumplir el proyecto.

7.1.2. Requerimientos no funcionales

Tabla 5

Tabla informativa de los requerimientos no funcionales del sistema.

ID	Nombre del requerimiento	Descripción
RNF1	Interoperabilidad	El sistema debe interactuar con las plataformas inmobiliarias: Finca Raíz, Metro cuadrado y Cien cuadras bajo el esquema: petición-respuesta. El intercambio de datos se debe llevar a cabo a través de peticiones (HTTP) y el formato de los datos debe ser JSON.
RNF2	Rendimiento	El sistema debe consultar las 3 plataformas inmobiliarias en paralelo para utilizar los recursos de hardware de forma eficiente y no afectar los tiempos de respuesta de cara al usuario.
RNF3	Usabilidad	El sistema debe ofrecer un formulario de búsqueda intuitivo que ofrezca la posibilidad al usuario de buscar una casa o apartamento por determinadas características y que al final éste pueda consultar el detalle de cada inmueble.
RNF4	Extensibilidad	El sistema debe tener la capacidad de admitir nuevos buscadores y poder ampliar fácilmente la búsqueda hacia otras plataformas inmobiliarias.

Fuente: Elaboración propia

Nota: La tabla presenta la información de los requerimientos funcionales que debe cumplir el proyecto.

8. Diseño

8.1. Creación del prototipo

Se realizó el diseño del prototipo de la aplicación en la herramienta de diagramación Lucidchart, aquí se definió la maqueta de como lucirá la aplicación en un entorno web y móvil, a continuación, se relacionan los diagramas realizados:

Entorno web: Se definieron dos vistas, en la vista principal se encuentra el buscador donde inicialmente se ingresarán los datos de ubicación, tipo de inmueble y tipo de venta o arriendo, al dar clic en el botón buscar se cargarán los nuevos filtros donde el usuario podrá o no filtrar por precio, estrato, cantidad de habitaciones, baños o parqueaderos seguido de la búsqueda realizada por house finder. Cada vez que se seleccione un filtro secundario esta búsqueda se actualizará con los parámetros indicados. Al dar clic en el botón más información se direccionará a la segunda vista.


Ilustración 6


Primera vista del prototipo web


The image shows a web browser window with the title 'HOUSE FINDER'. The search bar includes fields for 'Ubicación', 'Tipo Inmueble' (set to 'Casa - Apartamento'), and 'Venta' (set to 'En venta - Arriendo'), with a 'Buscar' button. The left sidebar contains filters for 'Valor y estado' (Precio, Estrato) and 'Tamaños y espacios' (Habitaciones, Baños, Parqueaderos, Área). The main content area displays three property listings, each with a photo, a platform logo, a title, price, details, and a 'Más información' button.

Ubicación	Tipo Inmueble	Venta	Buscar
<input type="text" value="Ubicación"/>	<input type="text" value="Casa - Apartamento"/>	<input type="text" value="En venta - Arriendo"/>	<input type="button" value="Buscar"/>

Valor y estado	Tamaños y espacios
Precio <input type="text" value="Precio desde COP"/> <input type="text" value="Precio hasta COP"/>	Habitaciones <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4+
Estrato <input checked="" type="radio"/> Todos <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6	Baños <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4+
	Parqueaderos <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4+
	Área <input type="text" value="Área desde (m2)"/> <input type="text" value="Área hasta (m2)"/>

LOGO DE PLATAFORMA: FINCA RAIZ
 <div> <p>Casa en Venta:</p> <p>\$430.000.000</p> <p>400 M2 - 6 Hab - 3 baños</p> <p>Suba - Bogotá</p> <input type="button" value="Más información"/> </div>

LOGO DE PLATAFORMA: METRO CUADRADO
 <div> <p>Apartamento en Venta:</p> <p>\$500.000.000</p> <p>100 M2 - 3 Hab - 2 baños</p> <p>Suba - Bogotá</p> <input type="button" value="Más información"/> </div>

LOGO DE PLATAFORMA: CIEN CUADRAS
 <div> <p>Casa en Venta:</p> <p>\$1200.000.000</p> <p>600 M2 - 5 Hab - 3 baños</p> <p>Suba - Bogotá</p> <input type="button" value="Más información"/> </div>

Fuente: Elaboración propia

Nota: Diagrama creado en la herramienta Lucidchart

Ilustración 7

segunda vista del prototipo web

Descripción general:

¡Casa en venta en Suba, 2do sector! Ubicación ideal: A solo 300 mts de la avenida Villavicencio y a 500 mts del COL (integración social). Amplio espacio: Lote de 140 mts² con 318 mts² construidos, distribuidos en 3 pisos. Precio: 400.000.000.00 Mct/e Distribución: 1er piso: Un apartamento de dos habitaciones, sala, comedor, baño, cocina y patio de ropas compartido, más una espaciosa habitación independiente con baño, cocina y patio de ropa compartido. 2do piso: Dos apartamentos con dos habitaciones, baño, sala, comedor, cocina y espacio para lavado. 3er piso: Dos apartamentos con dos habitaciones, baño, sala, comedor, cocina y espacio para lavado. 4to piso: Terraza de 12m x 14m para disfrutar de hermosas vistas. ¡Tu hogar ideal te espera en Arboleda Alta! Contáctanos para más información.

Simulador de gastos notariales		
Gastos compartidos		
Escrituración \$ xxxx.xxxx		
Gastos para el vendedor		
Escrituración \$ xxxx.xxxx		
Gastos para el comprador		
Escrituración : \$ xxxx.xxxx		
Impuesto Beneficencia: \$xxx.xxxx		
Impuesto registro: \$xxx.xxxx		

Fuente: Elaboración propia

Nota: Diagrama creado en la herramienta Lucidchart

En la segunda vista se podrá observar el detalle del inmueble, donde se mostrará la descripción general del mismo, contacto del vendedor y una sección en la que el usuario podrá ver los gastos asociados a la compra del inmueble.

Entorno Móvil: Se definieron tres vistas, en la vista principal se encuentra el buscador donde inicialmente se ingresaran los datos de ubicación, tipo de inmueble y tipo de venta o arriendo, al dar clic en el botón buscar se cargará un botón con los nuevos filtros (pero aparecerán en otra vista) al igual que el entorno web el usuario podrá o no filtrar por precio, estrato, cantidad de habitaciones, baños o parqueaderos seguido de las búsqueda realizada por house finder, cada vez que se seleccione un filtro secundario esta búsqueda se actualizará con los parámetros indicados. Al dar clic en el botón más información se direccionará a la segunda vista.

Ilustración 8

primera vista del prototipo móvil



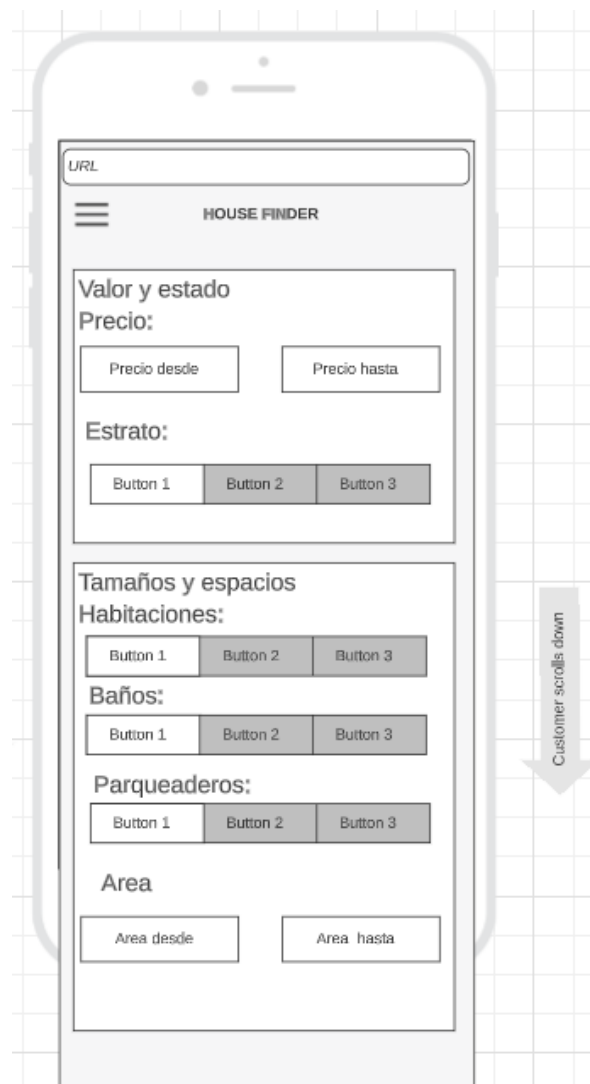
Fuente: Elaboración propia

Nota: Diagrama creado en la herramienta Lucidchart

Vista dos con los filtros:

Ilustración 9

Segunda vista del prototipo móvil



Fuente: Elaboración propia

Nota: Diagrama creado en la herramienta Lucidchart

Vista tres: Se podrá observar el detalle del inmueble, donde se brindará la descripción general del mismo, contacto con el vendedor y un simulador de gastos notariales que se deben asumir en el proceso de la compra.

Ilustración 10

Tercera vista del prototipo móvil



Fuente: Elaboración propia

Nota: Diagrama creado en la herramienta Lucidchart

9. Diseño y arquitectura

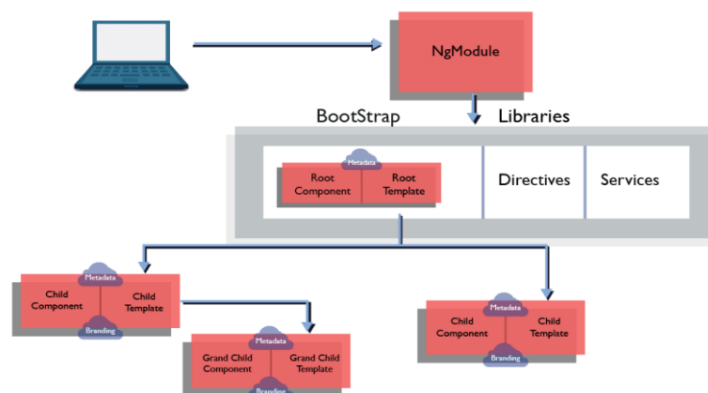
9.1. Arquitectura en angular

Como se mencionó en el punto 6 de herramientas a usar, Angular es el framework seleccionado para el diseño del frontend, este marco nos permite utilizar la arquitectura MVC o modelo vista controlador que nos ayuda a llevar un desacoplamiento entre la interfaz y la lógica del negocio. Las aplicaciones en angular están definidas por un conjunto de cajas, similar a

bloques de lego, que se pueden denominar módulos y estos puede asociar código funcional a sus componentes como servicios, para formar unidades funcionales.

Ilustración 11

Arquitectura de angular



Fuente: Documentación Microsoft ASPNET

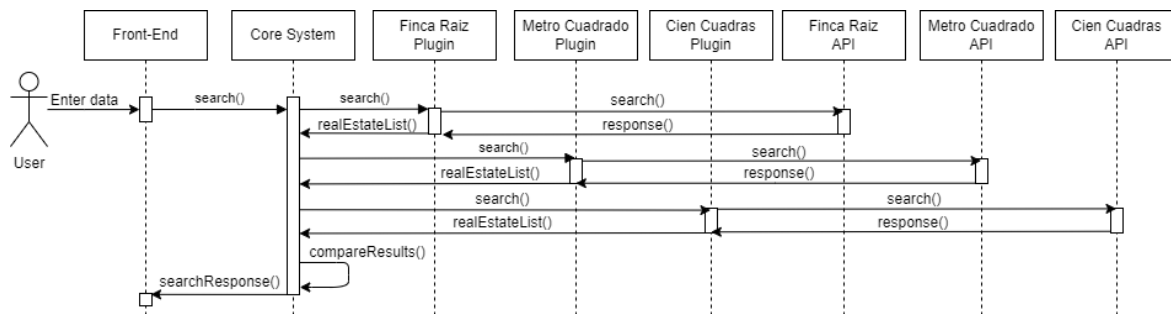
Nota: describe la arquitectura usada por el modelo MVC en .NET

Diagramas

Diagramas de Comportamiento:

Ilustración 12

Diagrama de Secuencia



Fuente: Elaboración Propia

Nota: Describe la interacción entre los objetos del sistema

Fuente: Elaboración Propia

Ilustración 14

Diagrama de Actividades

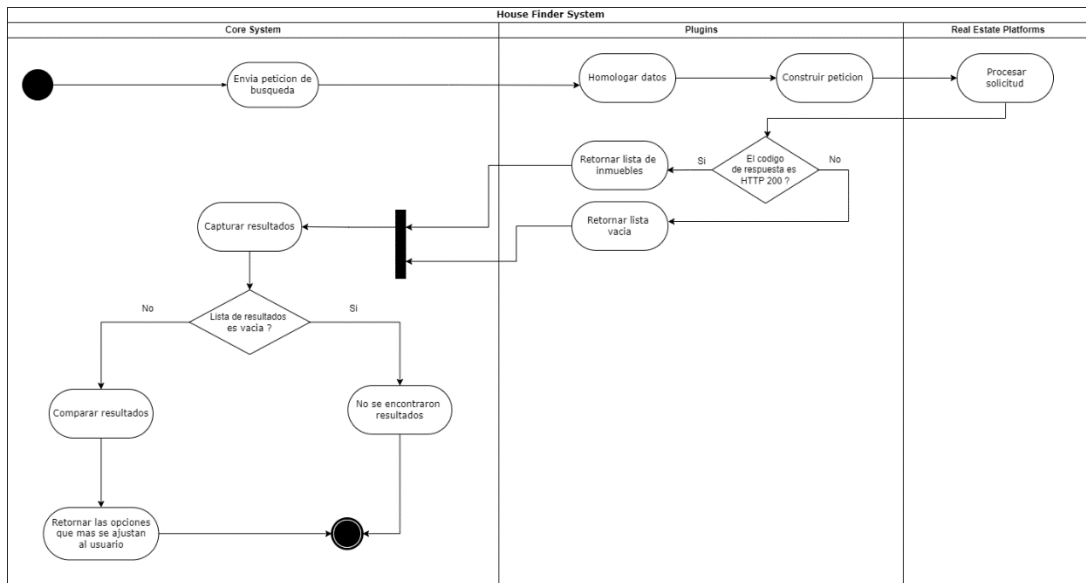
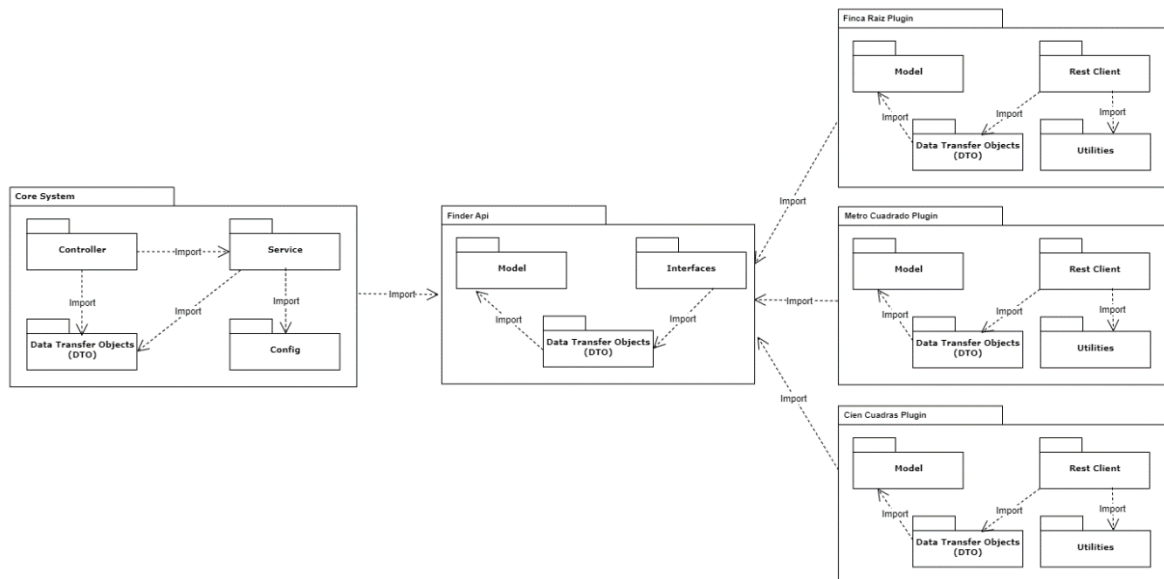


Ilustración 13

Diagramas de Estructura:

Nota: Muestra las actividades ejecutadas en el sistema

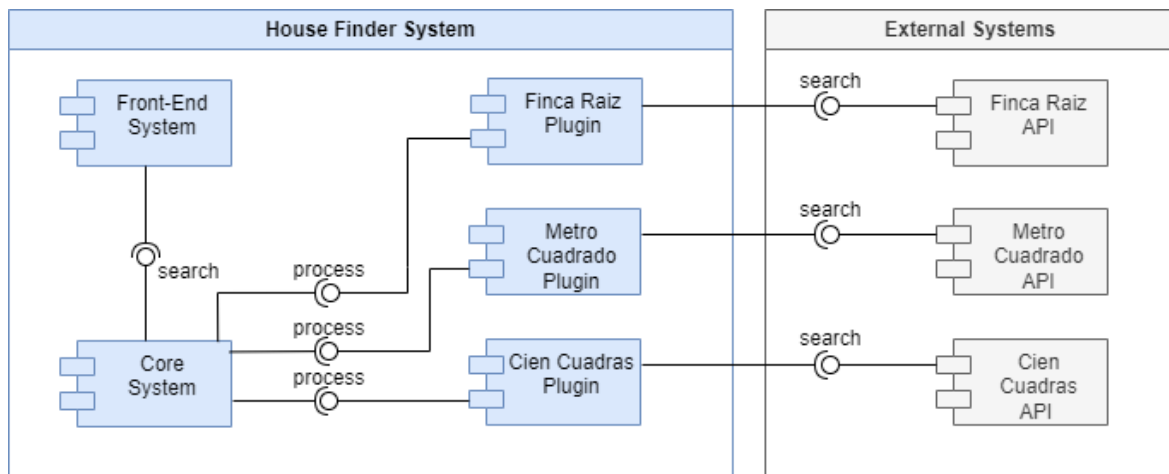


Fuente: Elaboración Propia

Nota: Indica las relaciones de dependencia entre los paquetes que forman la configuración de la aplicación.

Ilustración 15

Diagrama de Componentes



Fuente: Elaboración Propia

Nota: Permite visualizar la organización de los componentes del sistema y las relaciones de dependencia entre ellos.

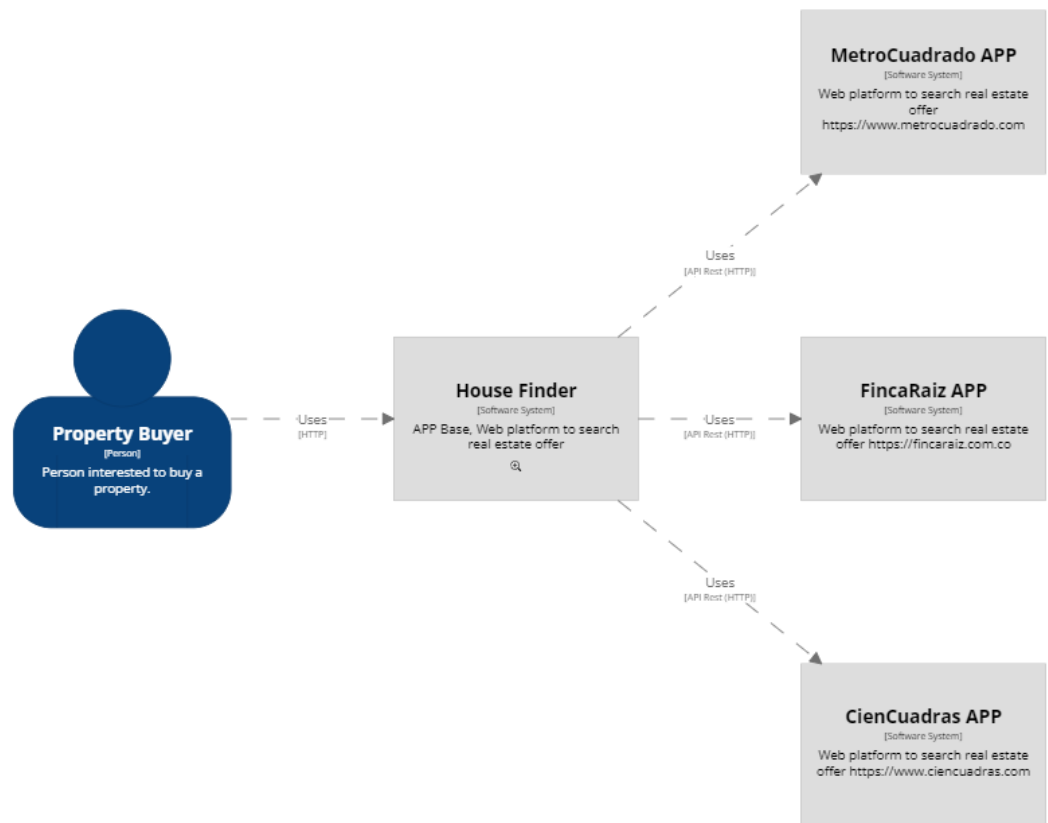
9.2. Arquitectura:

Se adoptó el modelo C4 para representar la arquitectura del proyecto y proporcionar diferentes niveles de abstracción:

Nivel 1

Ilustración 16

Nivel 1: Diagrama de contexto del sistema

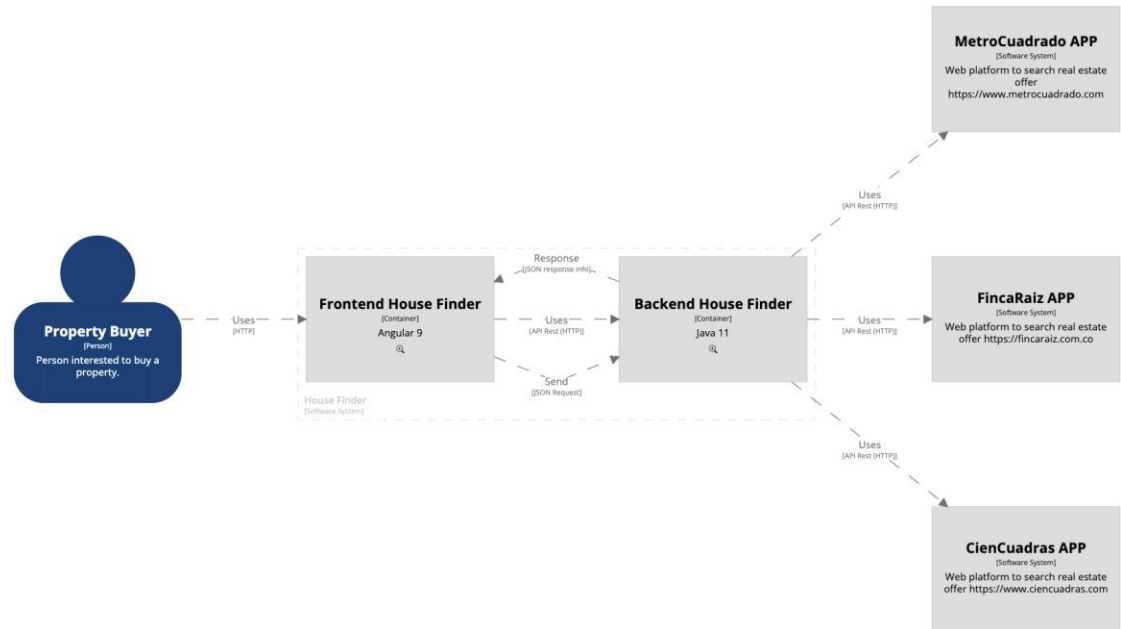


Fuente: Elaboración Propia, herramienta structurizr.com

Nota: Muestra el diagrama de contexto de House Finder

Nivel 2: Ilustración 17

Nivel 2: Diagrama de contenedores



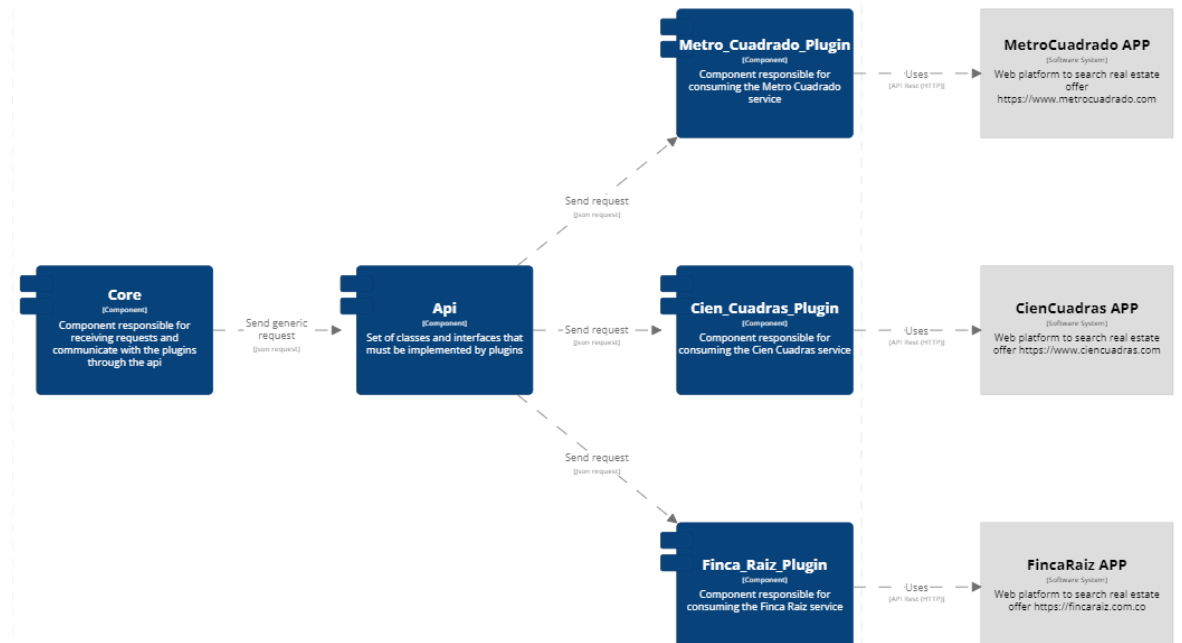
Fuente: Elaboración Propia, herramienta structurizr.com

Nota: Muestra el diagrama de contenedores de House Finder

Nivel 3:

Ilustración 18

Nivel 3: Diagrama de componentes de Back-end

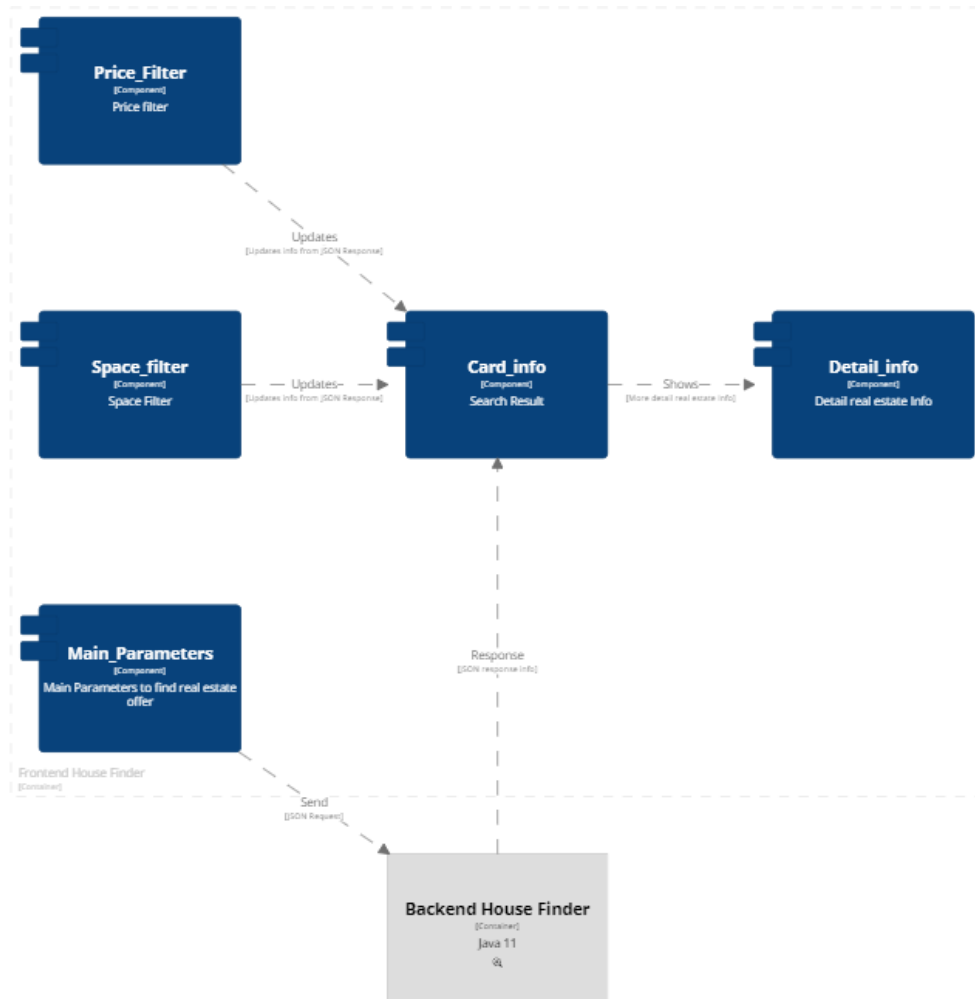


Fuente: Elaboración Propia, herramienta structurizr.com

Nota: Muestra el diagrama de componentes del backend de House Finder

Ilustración 19

Componentes de Front-end



Fuente: Elaboración Propia, herramienta structurizr.com

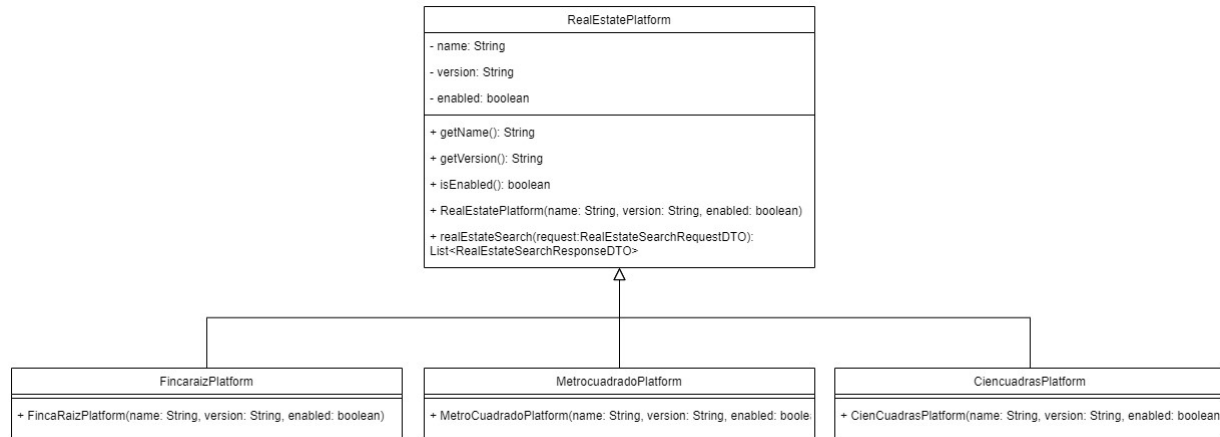
Nota: Muestra el diagrama de componentes del frontend de House

Finder

Nivel 4: Código

Ilustración 20

Nivel 4: Código

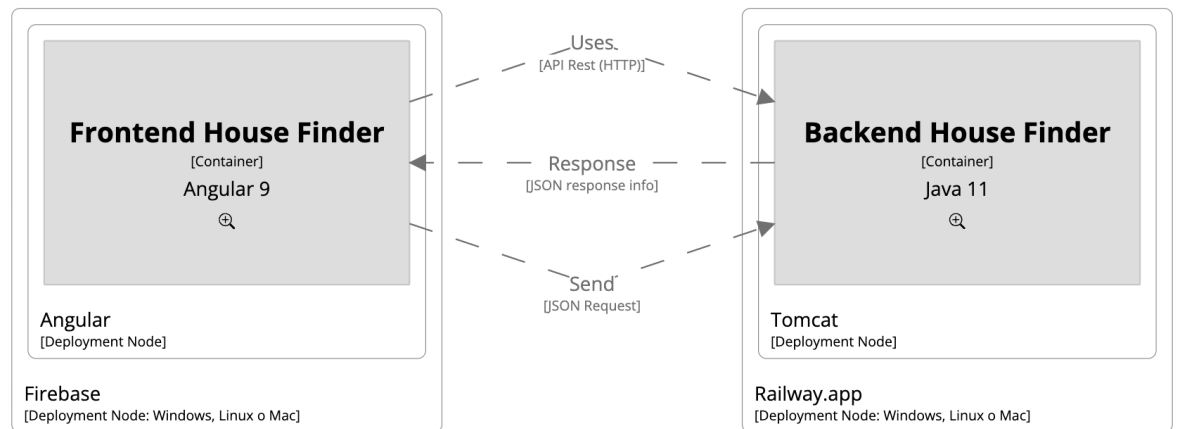


Fuente: Elaboración Propia, herramienta structurizr.com

Nota: Muestra el diagrama de clases de House Finder

Ilustración 21

Diagrama de despliegue



Fuente: Elaboración Propia, herramienta structurizr.com

Nota: Muestra el diagrama de despliegue de House Finder

9.3. Arquitectura de alto nivel:

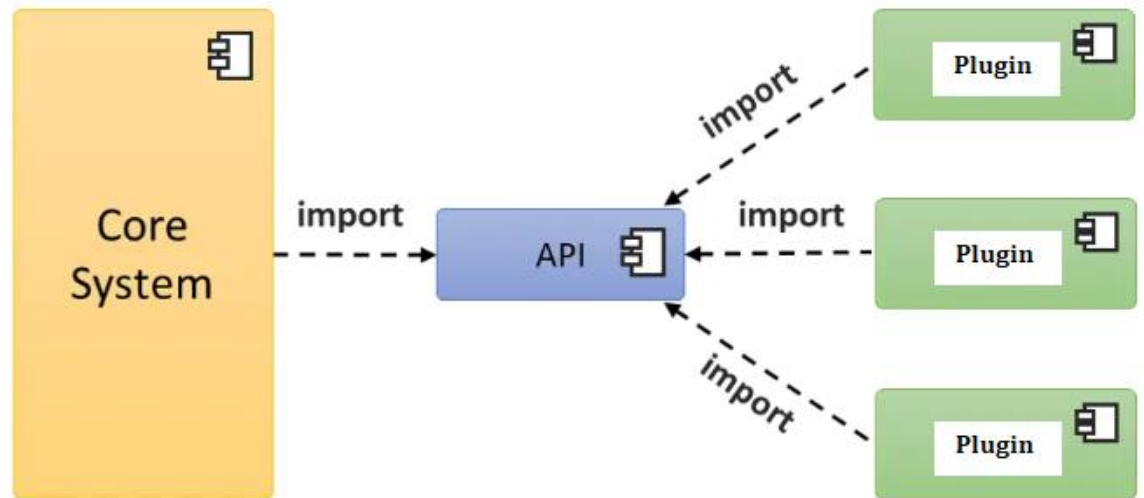
Teniendo en cuenta que House Finder interactuara con 3 plataformas inmobiliarias, el proyecto se implementara bajo una arquitectura de Plugins ya que esta permite crear una aplicación que se puede ampliar, permitiendo la incorporación de nuevas funciones mediante la adición de plugins. En este caso, cada plataforma inmobiliaria se manejará como un plugin, de esta forma se podrá extender la funcionalidad del Core System y si más adelante surge una nueva plataforma inmobiliaria esta podrá ser incorporada fácilmente al sistema sin necesidad de cambiar bruscamente el Core System, respetando el principio Open-closed.

En esta arquitectura se consideran los siguientes elementos:

- Core System: Sistema Central
- API: Conjunto de clases e interfaces que deben ser implementadas por cada Plugin
- Plugin: Módulo con una funcionalidad específica (para este caso: plataformas inmobiliarias).

Ilustración 22

Arquitectura de Microkernel



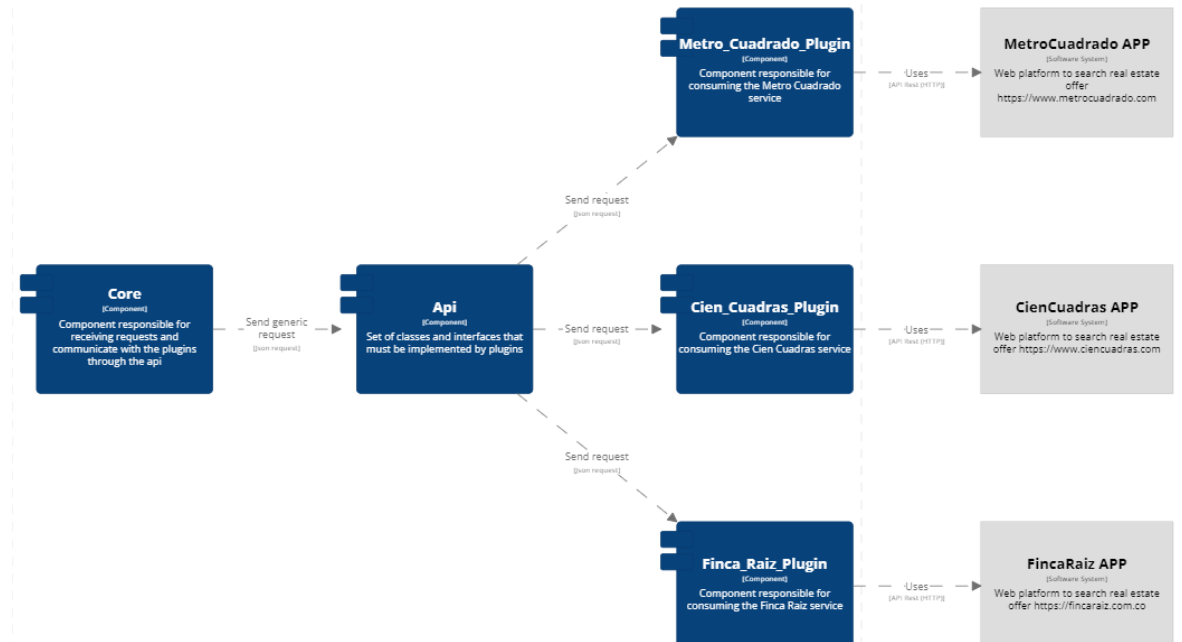
Fuente: [https://reactiveprogramming.io/blog/es/estilos-](https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/microkernel)

[arquitectonicos/microkernel](https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/microkernel) (12)

Nota: Permite visualizar los componentes de una arquitectura de plugins y la interacción entre ellos.

Ilustración 23

Arquitectura de Microkernel House Finder

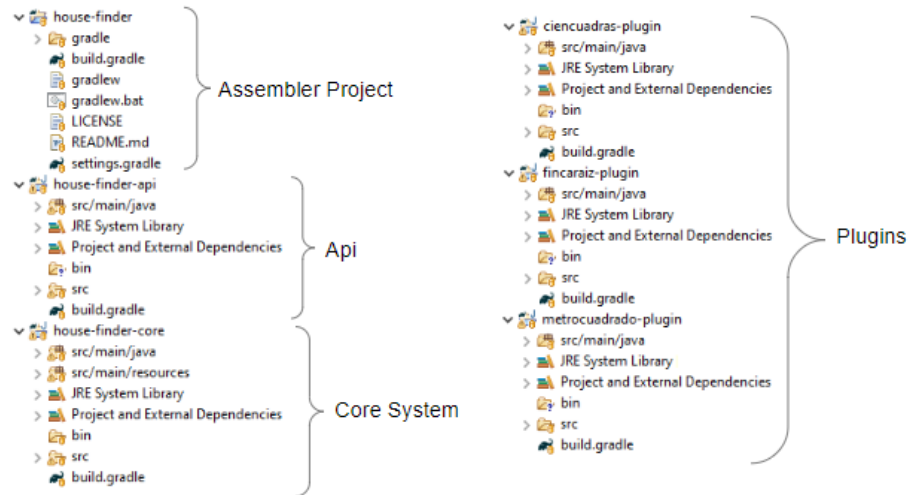


Fuente: Elaboración propia

Nota: Permite visualizar los componentes de House Finder y la interacción entre ellos.

Ilustración 24

Arquitectura de Microkernel House Finder - Implementación en Java



Fuente: Elaboración propia

Nota: Permite visualizar los diferentes subproyectos que componen la arquitectura de plugins

10. Construcción

10.1. Definir flujo de negocio:

En las tres plataformas de consulta (metro cuadrado, finca raíz y ciencuadras) se usaran sus servicios API para nuestro aplicativo House finder, cada vez que el usuario genere una nueva búsqueda, se activará el servicio rest a cada uno de los portales trayendo un listado de 30 publicaciones en cada uno de ellos, a nivel interno se realizará una validación dando prioridad al precio como parámetro de ordenamiento y presentando al usuario en la primera vista un listado de las 10 publicaciones más opcionadas según sus criterios ingresados.

Parámetros de búsqueda:

Se perfilaron los siguientes parámetros de búsqueda para utilizar en el aplicativo y en los servicios enviados a todos los portales:

Tabla 6*Tabla parámetros de búsqueda para House Finder.*

Nombre del campo	Descripción	Tipo de dato	Longitud	Etiqueta del dato
Ubicación	Campo en el cual se puede ingresar ciudad, barrio, localidad donde se quiere buscar el inmueble.	Text	255	location
Tipo de inmueble	Campo en el cual se selecciona que tipo de inmueble se va a buscar, ejemplo: Casa, Apartamento, Oficina, etc.	Text	255	typeProperty
Estado	Campo en el cual se puede seleccionar si el inmueble es nuevo o usado.	Text	255	stateProperty
Precio min	Campo en el cual se ingresa cual es el precio mínimo de búsqueda.	Int		startPrice
Precio max	Campo en el cual se ingresa cual es el precio máximo de búsqueda.	Int		endPrice
Habitaciones	Campo en el cual se selecciona el numero de habitaciones a buscar y filtrar	Int		room
baños	Campo en el cual se selecciona el numero de baños a buscar y filtrar	Int		amRoom
parqueaderos	Campo en el cual se selecciona el numero de parqueaderos a buscar y filtrar	Int		amParking
Estrato	Campo en el cual se selecciona el estrado que se va a buscar.	Int		socialEst
área min	Campo en el cual se ingresa cual es el área mínimo de búsqueda.	Int		starArea
área max	Campo en el cual se ingresa cual es el área máximo de búsqueda.	Int		endArea

Fuente: Elaboración Propia

Nota: La tabla describe el diccionario de datos de los parámetros usados en la aplicación.

11. Instalación y configuración

En este apartado se definirá el despliegue y configuración de frontend en angular, para desplegarlo inicialmente se ejecuta el comando `npm run build --prod`, para generar los archivos estáticos.

Ilustración 25

Comando para desplegar proyectos en angular

```
D:\Web\HouseFinder>npm run build --prod
npm WARN config production use --omit=dev instead.

> house-finder@0.0.0 build
> ng build

✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.
```

Initial Chunk Files	Names	Raw Size	Estimated Transfer Size
main.ee7d9c90bc75ab46.js	main	456.84 kB	101.02 kB
styles.bdd58c65057a942e.css	styles	67.59 kB	6.70 kB
polyfills.06d6d94ab5dd1d31.js	polyfills	33.02 kB	10.67 kB
runtime.82f37287db40750d.js	runtime	900 bytes	516 bytes
	Initial Total	558.33 kB	118.89 kB

```
Build at: 2023-11-09T18:17:45.033Z - Hash: 4fd68249c50ac0a2 - Time: 52416ms
Warning: bundle initial exceeded maximum budget. Budget 500.00 kB was not met by 58.33 kB with a total of 558.33 kB.
```

Fuente: Elaboración Propia

Nota: La ilustración describe el paso inicial para desplegar un proyecto en angular.

Para cargar la información en Firebase, se inicia sesión con el comando firebase login en la consola de angular, esto lleva a una ventana emergente en el navegador donde se selecciona la cuenta con la que se registró en la herramienta y se evidencia el mensaje de login exitoso:

Ilustración 26

Segundo paso de despliegue en angular

```
D:\Web\HouseFinder>firebase login
i Firebase optionally collects CLI and Emulator Suite usage and error reporting information to help improve our products. Data is collected in accordance with Google's privacy policy (https://policies.google.com/privacy) and is not used to identify you.

? Allow Firebase to collect CLI and Emulator Suite usage and error reporting information? No

Visit this URL on this device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=563584335869-fgrhgmd47bqnekij5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&response_type=code&state=1019848394&redirect_uri=http%3A%2F%2Flocalhost%3A9005

Waiting for authentication...
+ Success! Logged in as fernanda.aldana08@gmail.com
```

Fuente: Elaboración Propia

Nota: La ilustración describe el segundo paso para desplegar un proyecto en angular.

Para iniciar el proyecto se ingresa el comando firebase init hosting en la consola de angular previamente en el navegador se había creado el nuevo proyecto en el apartado de hosting y es el proyecto que se selecciona para este proceso:

Ilustración 27

Comando de despliegue en firebase

```
D:\Web\HouseFinder>firebase init hosting

#####
##          ##          ##          ##          ##          ##          ##          ##          ##          ##          ##
#####
##          ##          ##          ##          ##          ##          ##          ##          ##          ##          ##
##          ##          ##          ##          ##          ##          ##          ##          ##          ##          ##
##          ##          ##          ##          ##          ##          ##          ##          ##          ##          ##

You're about to initialize a Firebase project in this directory:

  D:\Web\HouseFinder

? Are you ready to proceed? Yes

=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: Use an existing project
? Select a default Firebase project for this directory: housefinder-77810 (HouseFinder)
i Using project housefinder-77810 (HouseFinder)

=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? dist/angular-keep
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? No
+ Wrote dist/angular-keep/index.html

i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...

+ Firebase initialization complete!
```

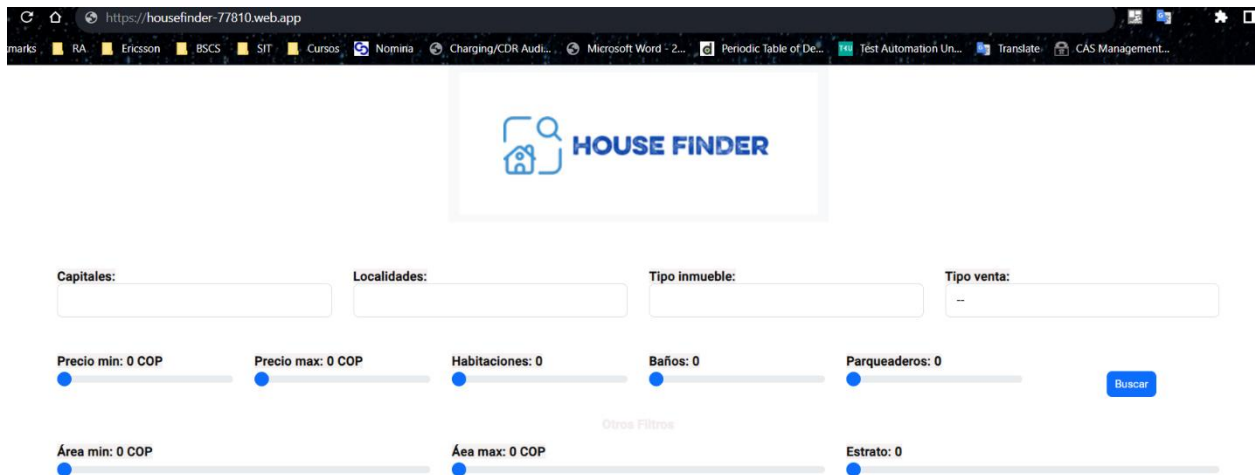
Fuente: Elaboración Propia

Nota: La ilustración describe el segundo paso para desplegar un proyecto en angular.

Se puede visualizar la aplicación desplegada en el servidor hosting de firebase

Ilustración 28

Aplicación desplegada en firebase



Fuente: Elaboración Propia

Nota: La ilustración muestra la aplicación desplegada en el hosting de firebase.

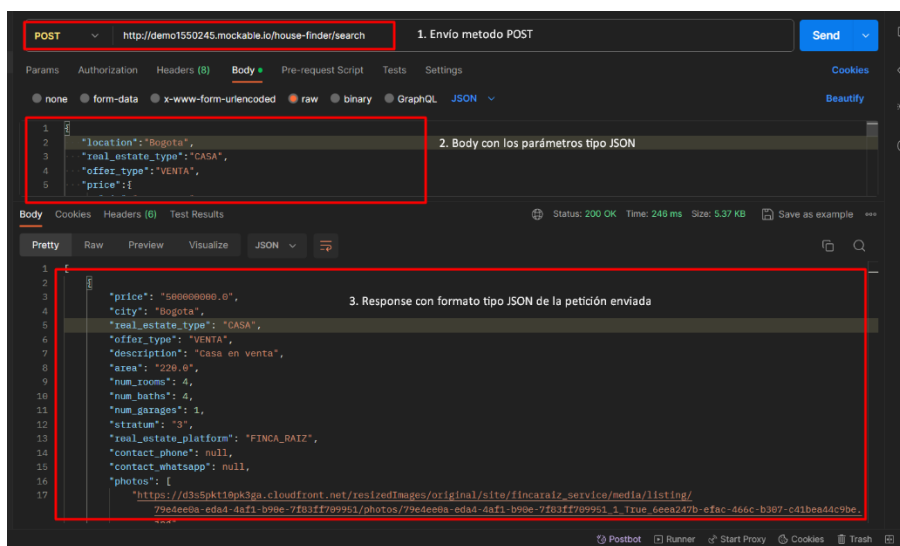
12. Pruebas y calidad de código

12.1. Pruebas de caja negra:

Se utilizó la herramienta Postman para realizar la prueba de los servicios API Rest, se valida que al enviar el servicio post con los parámetros de búsqueda se reciba una respuesta con formato JSON:

Ilustración 29

Resultado pruebas caja negra en postman



Fuente: Elaboración Propia

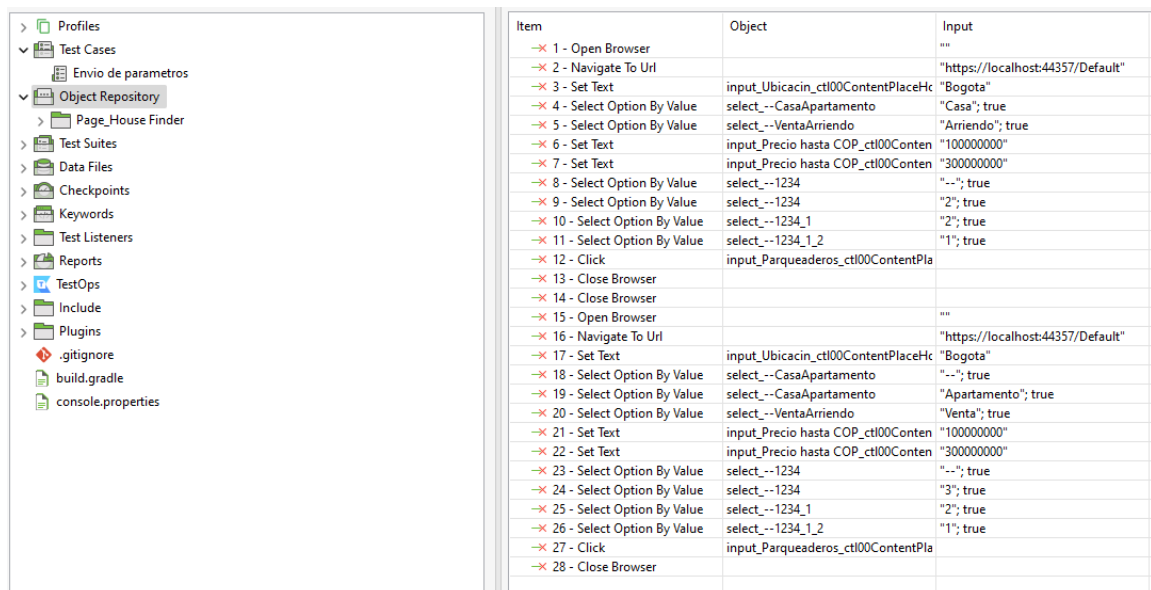
Nota: La imagen muestra el resultado de las pruebas de caja negra aplicadas a la herramienta.

12.2. Pruebas funcionales:

Para las pruebas funcionales de House Finder se utilizó la herramienta Katalon Studio que permite grabar las interacciones en la página para luego ejecutarlas de forma automatizada en el navegador Google Chrome.

Ilustración 30

Pasos automatizados en Katalon Studio



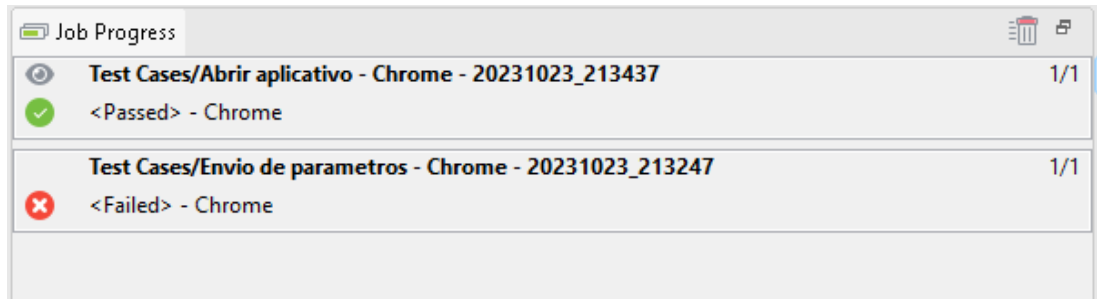
Item	Object	Input
1 - Open Browser		""
2 - Navigate To Url		"https://localhost:44357/Default"
3 - Set Text	input_Ubicacin_ctl00ContentPlaceHc	"Bogota"
4 - Select Option By Value	select_--CasaApartamento	"Casa"; true
5 - Select Option By Value	select_--VentaArriendo	"Arriendo"; true
6 - Set Text	input_Precio hasta COP_ctl00Conten	"100000000"
7 - Set Text	input_Precio hasta COP_ctl00Conten	"300000000"
8 - Select Option By Value	select_--1234	"-"; true
9 - Select Option By Value	select_--1234	"2"; true
10 - Select Option By Value	select_--1234_1	"2"; true
11 - Select Option By Value	select_--1234_1_2	"1"; true
12 - Click	input_Parqueaderos_ctl00ContentPla	
13 - Close Browser		
14 - Close Browser		
15 - Open Browser		""
16 - Navigate To Url		"https://localhost:44357/Default"
17 - Set Text	input_Ubicacin_ctl00ContentPlaceHc	"Bogota"
18 - Select Option By Value	select_--CasaApartamento	"-"; true
19 - Select Option By Value	select_--CasaApartamento	"Apartamento"; true
20 - Select Option By Value	select_--VentaArriendo	"Venta"; true
21 - Set Text	input_Precio hasta COP_ctl00Conten	"100000000"
22 - Set Text	input_Precio hasta COP_ctl00Conten	"300000000"
23 - Select Option By Value	select_--1234	"-"; true
24 - Select Option By Value	select_--1234	"3"; true
25 - Select Option By Value	select_--1234_1	"2"; true
26 - Select Option By Value	select_--1234_1_2	"1"; true
27 - Click	input_Parqueaderos_ctl00ContentPla	
28 - Close Browser		




Fuente: Elaboración Propia

Nota: La imagen muestra los pasos grabados para la automatización de pruebas.

Ilustración 31

Pasos automatizados en Katalon Studio



Job Progress	
 Test Cases/Abrir aplicativo - Chrome - 20231023_213437	1/1
 <Passed> - Chrome	
Test Cases/Envio de parametros - Chrome - 20231023_213247	1/1
 <Failed> - Chrome	

Fuente: Elaboración Propia

Nota: La imagen muestra el resultado de las pruebas automatizadas.

13. Conclusiones

- En el transcurso de este trabajo de grado, se ha llevado a cabo un extenso proceso de desarrollo e implementación de la aplicación web "House Finder," cuyo propósito fundamental es centralizar y simplificar la búsqueda de inmuebles en Colombia. Este proyecto ofrece a los usuarios una plataforma accesible y eficiente para encontrar propiedades de manera más rápida y efectiva.
- La creación de "House Finder" responde a la necesidad creciente en Colombia de contar con una herramienta que unifique la información dispersa en múltiples portales y agilice el proceso de búsqueda de inmuebles, todo ello diseñado para mejorar la experiencia del usuario.
- Con el desarrollo del proyecto se logró agilizar el proceso de búsqueda de inmuebles y mejorar la experiencia del usuario.
- La información brindada por las plataformas inmobiliarias: Fincaraiz, Metrocuadrado y Ciencuadras fue de vital importancia para seleccionar las ofertas que más se ajustaran a las necesidades del usuario.

14. Referencias

1. Similarweb. (2023, 1 noviembre). Sitios web inmobiliaria más visitados en Colombia. Similarweb. <https://www.similarweb.com/es/top-websites/colombia/business-and-consumer-services/real-estate/>
2. (Masse, 2011) Masse, M. (2011). REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces. O'Reilly Media, Inc.
3. (Benzell et al., 2019) (3)
4. <https://www.habitatbogota.gov.co/prensa/noticias/pensando-comprar-vivienda-estos-documentos-son-los-debes-pedir>
5. Osorio-Sanabria et al. (2021) (4)
6. Osorio-Sanabria, M. A., Amaya-Fernández, F., & González-Zabala, M. P. (2021). Políticas, normas y estrategias que fomentan los datos abiertos en Colombia: un análisis de literatura. *Revista virtual Universidad Católica del Norte*, 62, 155-188. <https://doi.org/10.35575/rvucn.n62a7>
7. Garriga-Portolà, M. (2011). ¿ Datos abiertos? Sí, pero de forma sostenible. *Profesional de la Información*, 20(3), 298-303.
8. Trello (2023), recuperado de <https://trello.com/es/tour>

9. IBM (2023), Ventajas de Java, recuperado de <https://www.ibm.com/docs/es/aix/7.3?topic=monitoring-advantages-java>
10. Arquitectura de Plugins o Microkernel, Blancarte Oscar, Arquitectura de Microkernel, (2023). Reactive Programming. <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/microkernel>
11. ¿Qué es Kanban?, (2023), recuperado de <https://www.nimblework.com/es/kanban/que-es-kanban/>
12. Diagrama <https://www.linkedin.com/pulse/arquitectura-de-plugins-o-microkernel-ariel-alejandro-wagner/?originalSubdomain=es>
13. González Tocarruncho, J. F. (2022). El derecho de acceso a la información pública en Colombia: una mirada desde el cumplimiento de la Ley 1712 de 2014.
14. Cifuentes Muñoz, E. (1997). El hábeas data en Colombia. Derecho PUCP, 51, 115.