

SerralInnova



Alejandro Ruiz Chilet

Índice

1.- Análisis.....	3
.....	3
1.1.- Descripción del problema y del formulario asignado (objetivo, usuario objetivo y contexto).....	3
1.2.- Requisitos funcionales qué debe cumplir el formulario asignado.....	3
1.3.- Modelo entidad-relación.....	6
1.4.- Diccionario de datos: campos, tipo, obligatoriedad, validación y ejemplo de valor.....	7
2.- Diseño UI.....	9
2.1.- Mockups del diseño final de las pantallas.....	9
2.2.- Mapa de pantallas/flujo de navegación.....	11
2.3.- Breve guía de estilo: paleta de colores, tipografías, tamaños y estilos de componentes.....	12
3.- Desarrollo.....	15
3.1.- Desarrollar el formulario en base al análisis realizado.....	15
3.2.- Persistencia los datos registrados en el formulario.....	16
3.3.- Validación de campos obligatorios, formatos, rangos. Mostrar errores de forma clara junto al campo y evitar registrar información errónea o inválida.....	16
3.4.- Navegación clara entre las pantallas y gestión de estados (provider).....	16
3.5 – Estructura del proyecto conforme se ha realizado en el resto de proyectos (lib/models, lib/routes, lib/widgets, lib/provider, lib/pages).....	17
3.6.- Diseño de la interfaz.....	17
4.- Pruebas.....	18

1.- Análisis

1.1.- Descripción del problema y del formulario asignado (objetivo, usuario objetivo y contexto).

El problema que aborda este proyecto es la dificultad que tienen muchas personas para encontrar viviendas que cumplan criterios de sostenibilidad y accesibilidad mediante los buscadores inmobiliarios tradicionales, ya que estos no suelen ofrecer filtros específicos para colectivos vulnerables ni para características medioambientales.

El objetivo del formulario es permitir a los usuarios buscar y filtrar viviendas de manera sencilla, teniendo en cuenta criterios de eficiencia energética, sostenibilidad y adaptación a diferentes necesidades sociales, facilitando así el acceso a la vivienda a colectivos como personas con discapacidad, personas mayores, familias, jóvenes y otros grupos en situación de vulnerabilidad.

El usuario objetivo es cualquier persona interesada en alquilar o comprar una vivienda sostenible, especialmente aquellas que pertenecen a colectivos prioritarios o que requieren viviendas adaptadas. También puede ser utilizado por profesionales del sector inmobiliario para asesorar a clientes con necesidades específicas.

El contexto de uso del formulario es una plataforma web o móvil de la inmobiliaria SERRAINNOVA, donde el usuario accede para buscar viviendas disponibles aplicando distintos filtros y visualizando los resultados que mejor se ajustan a sus necesidades económicas, sociales y medioambientales.

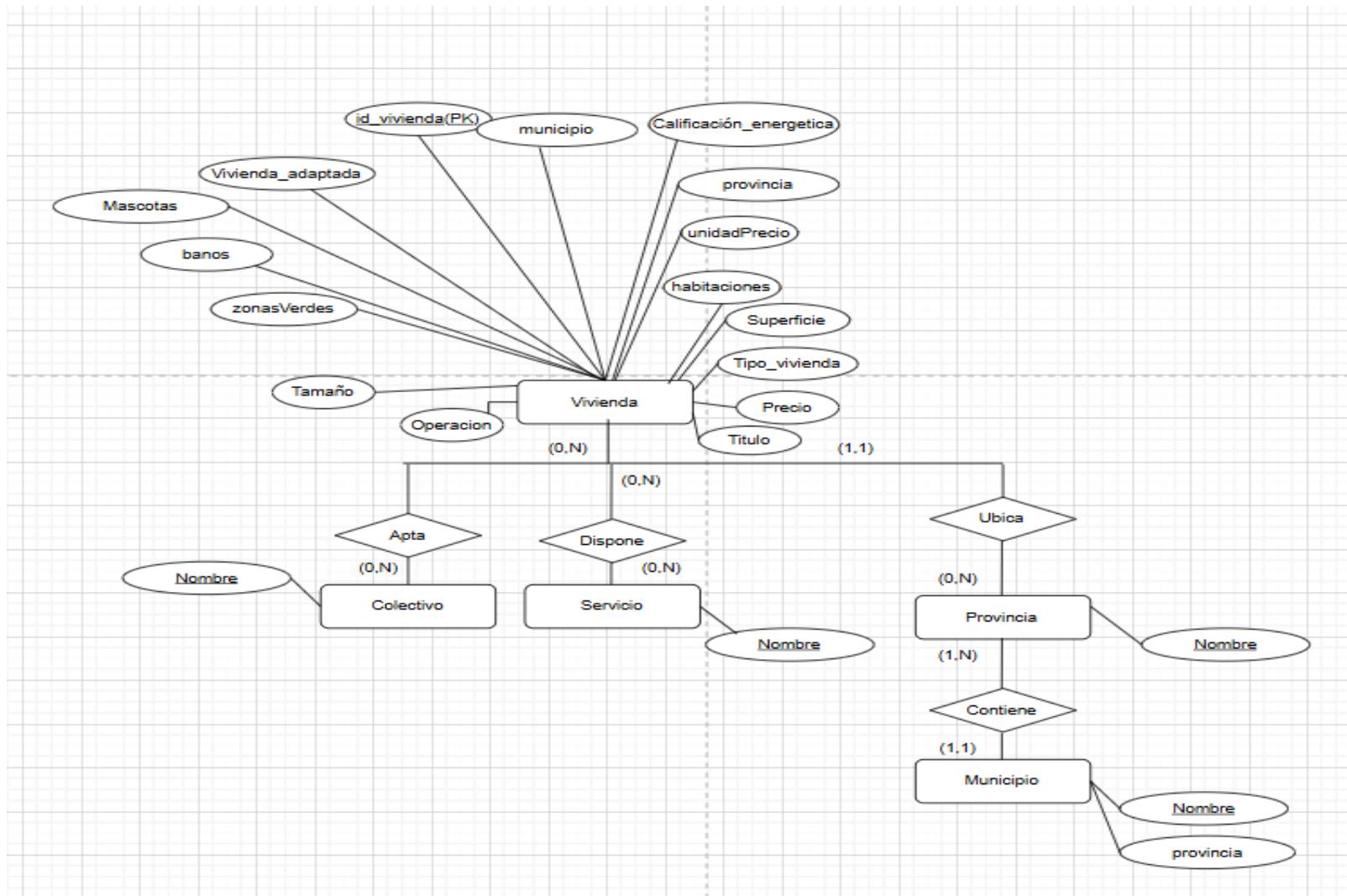
1.2.- Requisitos funcionales qué debe cumplir el formulario asignado.

1. El sistema debe permitir al usuario seleccionar el tipo de operación, ya sea alquiler o compra
2. El sistema debe disponer de un menú de filtros accesible desde la interfaz principal.
3. El sistema debe permitir aplicar filtros básicos, entre ellos:
 - Rango de precio (mínimo y máximo)
 - Zona (provincia y municipio)
 - Tamaño de la vivienda (superficie en m²)
 - Extras (terraza, garaje, trastero)
4. El sistema debe permitir aplicar filtros avanzados organizados por categorías:
 - **Sostenibilidad**
 - Calificación energética
 - Tipo de energía renovable

- Certificación ambiental
- Nivel estimado de emisiones de CO₂
- **Accesibilidad y características sociales**
 - Vivienda adaptada (sí/no)
 - Colectivos prioritarios:
 - Personas mayores
 - Jóvenes
 - Familias
 - Personas con discapacidad
 - Servicios cercanos:
 - Centros de salud
 - Transporte público
 - Centros educativos
 - Vivienda apta para mascotas (sí/no)
 - Zonas verdes cercanas (sí/no)
- 5. El sistema debe permitir seleccionar características generales de la vivienda:
 - Tipo de vivienda (piso, casa, estudio)
 - Número de habitaciones
 - Número de baños
 - Zonas comunes:
 - Terraza
 - Garaje
 - Trastero
- 6. El sistema debe permitir filtrar por condiciones económicas:
 - Precio de venta o alquiler
 - Inclusión o no de gastos
 - Posibilidad de ayudas o subvenciones
 - Tipo de régimen (libre, protegido, alquiler social)

7. El sistema debe permitir ejecutar la búsqueda mediante un botón de acción.
8. El sistema debe mostrar un listado de resultados que cumpla con los filtros seleccionados.
9. El sistema debe permitir modificar los filtros y actualizar los resultados sin perder la búsqueda anterior.

1.3.- Modelo entidad-relación



El diagrama entidad-relación define la estructura de datos del sistema de búsqueda de viviendas de SERRAINNOVA. La entidad principal es Vivienda, que almacena la información básica del inmueble incluyendo características generales, económicas, de accesibilidad y sostenibilidad, así como su localización.

Las características relacionadas con sostenibilidad y accesibilidad (como calificación energética, emisiones de CO₂, vivienda adaptada, mascotas o zonas verdes) se almacenan directamente como atributos de la entidad Vivienda, simplificando el modelo y adaptándolo a una estructura de datos no relacional.

Los colectivos prioritarios y los **servicios cercanos** se gestionan mediante relaciones de muchos a muchos entre Vivienda y las entidades Colectivo y Servicio, permitiendo que una vivienda esté asociada a varios colectivos y servicios, y viceversa.

La ubicación se organiza mediante las entidades Provincia y Municipio, facilitando el filtrado geográfico.

1.4.- Diccionario de datos: campos, tipo, obligatoriedad, validación y ejemplo de valor

Vivienda:

Campo	Tipo	Obligatorio	Validación	Ejemplo	Mensaje de error
id_vivienda	Texto	Sí	Campo relleno	0IZ4xkPS2...	-
titulo	Texto	Sí		Casa en alquiler en Alicante	
superficie	Decimal	Sí	> 0	85.5	Superficie incorrecta
habitaciones	Entero	Sí	≥ 0	3	Valor no válido
banos	Entero	No	≥ 0	2	Valor no válido
precio	Decimal	Sí	> 0	750	Precio incorrecto
municipio	Texto	Sí	Campo especificado	Torrent	Municipio no válido
provincia	Texto	Sí	Campo especificado	Valencia	Provincia no válida
operacion	Texto	Sí	Campo especificado	Alquilar	Operación no elegida
unidadPrecio	Texto	Sí			
imageUrl	Texto	No			

Sostenibilidad:

Campo	Tipo	Obligatorio	Validación	Ejemplo	Mensaje de error
calificacion_energetica	Texto	Sí	A-G	A	
emisiones	Decimal	No	≥ 0	25.4	

Accesible:

Campo	Tipo	Obligatorio	Validación	Ejemplo	Mensaje de error
vivienda_adaptada	Booleano	Sí	Sí/No	Sí	Valor incorrecto
mascotas	Booleano	Sí	Sí/No	No	Valor incorrecto
zonas_verdes	Booleano	Sí	Sí/No	Sí	Valor incorrecto

Colectivo:

Campo	Tipo	Obligatorio	Validación	Ejemplo	Mensaje de error
nombre	Texto	Sí	Lista cerrada	Mayores	Colectivo no válido

Servicio:

Campo	Tipo	Obligatorio	Validación	Ejemplo	Mensaje de error
nombre	Texto	Sí	Lista cerrada	Centro salud	Servicio no válido

Provincia:

Campo	Tipo	Obligatorio	Validación	Ejemplo	Mensaje de error
nombre	Texto	Sí	Único	Valencia	Provincia no válida

Municipio:

Campo	Tipo	Obligatorio	Validación	Ejemplo	Mensaje de error
nombre	Texto	Sí	Único	Torrent	Municipio no válido

2.- Diseño UI

2.1.- Mockups del diseño final de las pantallas

Mockup de inicio:



Este es mockup de la página principal, donde el usuario dispone de una serie de filtros rápidos, donde puede elegir el tipo de operación, la ubicación por provincia y municipio, y también puede realizar la búsqueda por precio, superficie, habitaciones y extras. En la esquina superior derecha se encuentra un botón, el cuál abre una pantalla para los filtros avanzados, útiles para el usuario que quiera realizar una búsqueda más específica.

Mockup 2 – Filtros avanzados:

Mockup 2 - Filtros avanzados

Filtros avanzados

Sostenibilidad:

- Calificación energética (A-G)
 - Tipo de energía ▾
 - Certificación ▾
- Emisiones CO₂:

Accesibilidad:

- Vivienda adaptada
- Mascotas permitidas
- Zonas verdes cercanas

Colectivos prioritarios:

- Mayores
- Jóvenes
- Familias
- Discapacidad

Servicios cercanos:

- Centros de salud
- Transporte público
- Centros educativos

Características:

- Tipo vivienda ▾
- Habitaciones: --- ▾
- Baños: --- ▾
- Terraza
- Garaje
- Trastero

Economía:

- Precio:
Min. Max
- Gastos incluidos
- Ayudas
- Régimen ▾

Aplicar filtros

Este mockup representa la pantalla de filtros avanzados, donde el usuario puede personalizar la búsqueda mediante criterios de sostenibilidad, accesibilidad, colectivos prioritarios, servicios cercanos, características y economía. Se han utilizado distintos componentes como selectores, checks o interruptores para facilitar la interacción con el usuario y mejorar su experiencia.

Mockup 3 – Listado de resultados:

En este se mostrará la lista de resultados tras la búsqueda, donde aparecerán los detalles de la vivienda en la parte central, con paginación inferior para ver más imágenes. Además mostrará los filtros aplicados en la parte derecha.

2.2.- Mapa de pantallas/flujo de navegación

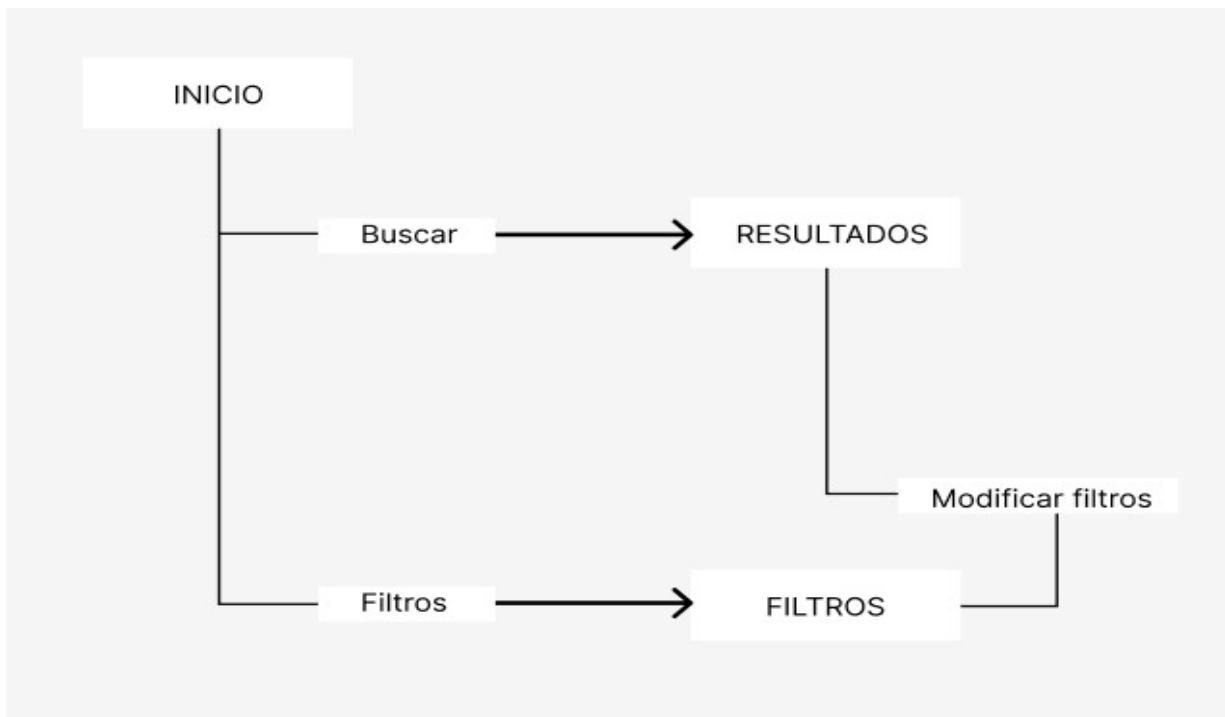
El flujo de navegación empieza en la pantalla de inicio, donde el usuario selecciona el tipo de operación, la zona de búsqueda y los filtros rápidos que deseé. Desde esta pantalla puede realizar una búsqueda o acceder a los filtros avanzados.

Si el usuario pulsa en filtros, accede a la pantalla de inicio, donde puede personalizar la búsqueda según criterios de sostenibilidad, accesibilidad, colectivos prioritarios, características de la vivienda y condiciones económicas. Al pulsar el botón “Aplicar filtros”, se muestran los resultados.

La pantalla de resultados presenta un listado de viviendas que cumplen los criterios seleccionados. Desde aquí, el usuario puede modificar los filtros o pulsar en “Ver detalles” para consultar la información completa de una vivienda.

Al seleccionar una vivienda, se accede a la pantalla de detalle, donde se muestra toda la información del inmueble organizada por secciones. Desde esta pantalla el usuario puede volver al listado.

2.3.- Breve guía de estilo: paleta de colores, tipografías, tamaños y estilos de componentes



La interfaz SerraInnova se ha diseñado siguiendo una línea visual limpia, moderna y coherente con los valores de sostenibilidad e inclusión social.

Paleta de colores:

- **Verde (#90EE90):** se utiliza para botones principales y elementos destacados. Representa sostenibilidad y compromiso ambiental.
- **Gris claro (#D3D3D3):** color de fondo principal para mejorar la legibilidad.
- **Blanco (#FFFFFF):** fondos de tarjetas y componentes de UI.
- **Negro (#000000):** texto principal, secundario y elementos informativos.

Tipografía:

- **Fuente principal:** Inter
 - Títulos: Inter Bold
 - Texto: Inter Regular
- **Tamaños:**
 - Títulos principales: 16-24 px
 - Texto normal: 12px

Componentes UI:

Botones

- Color: Verde principal
- Texto: Blanco
- Bordes redondeados
- Estados:
 - Hover: verde más oscuro
 - Disabled: gris

Campos de texto

- Fondo blanco
- Borde gris claro
- Placeholder en gris
- Focus: borde verde

Selectores (dropdown)

- Fondo blanco
- Flecha desplegable
- Opciones con hover verde claro

Checkbox

- Cuadro blanco
- Marcado en verde
- Usados para:
 - Colectivos
 - Servicios
 - Extras

Switch (interruptor)

- OFF: gris
- ON: verde
- Usador para:
 - Accesibilidad
 - Mascotas
 - Zonas verdes

Tarjetas

- Fondo blanco
- Bordes redondeados
- Separación clara entre elementos

Diseño responsive

- Adaptable a desktop
- Columnas en desktop
- Diseño vertical en móvil

Objetivo visual

- Interfaz clara
- Fácil de usar
- Accesible
- Enfocada en sostenibilidad

3.- Desarrollo

3.1.- Desarrollar el formulario en base al análisis realizado

SerraInnova es una aplicación desarrollada con Flutter cuyo objetivo es facilitar la búsqueda de viviendas sostenibles mediante un sistema de filtrado claro e intuitivo.

La aplicación permite al usuario buscar viviendas según distintos criterios como tipo de operación (compra o alquiler), ubicación, precio, superficie, accesibilidad y sostenibilidad.

El sistema está pensado para un uso principalmente web, aunque mantiene compatibilidad con dispositivos móviles gracias al diseño responsive de Flutter.

Formularios: home_page.dart

El formulario principal de búsqueda se encuentra en la pantalla inicial (HomePage).

En esta pantalla, el usuario puede seleccionar:

- Tipo de operación: comprar o alquilar.
- Ubicación: provincia y municipio.
- Filtros rápidos: precio, superficie, número de habitaciones y extras.

Los filtros rápidos se muestran mediante botones que abren diálogos modales (AlertDialog), permitiendo ajustar los valores sin abandonar la pantalla principal.

filtros_page.dart:

La aplicación incluye una pantalla de filtros avanzados accesible desde el menú superior.

En esta pantalla se agrupan los filtros en distintas secciones:

- Sostenibilidad (calificación energética, emisiones de CO₂, tipo de energía).
- Accesibilidad (vivienda adaptada, mascotas permitidas, zonas verdes).
- Colectivos prioritarios.
- Servicios cercanos.
- Características de la vivienda.
- Economía.

El uso de SingleChildScrollView permite mostrar una gran cantidad de opciones sin comprometer la usabilidad.

3.2.- Persistencia los datos registrados en el formulario

Los datos de las viviendas se almacenan en Firebase Firestore, utilizando una colección llamada viviendas.

Cada documento representa una vivienda e incluye información como: precio, ubicación, características, emisiones de CO₂, imagen asociada mediante una URL., las cuáles se recogen de un banco público de imágenes de viviendas.

Para facilitar el desarrollo y las pruebas, se ha implementado un sistema de carga inicial de datos (seed), el cuál se encuentra en la carpeta /dev, que genera varias viviendas de ejemplo.

3.3.- Validación de campos obligatorios, formatos, rangos. Mostrar errores de forma clara junto al campo y evitar registrar información errónea o inválida

Además, este formulario incorpora validaciones básicas para garantizar una experiencia de usuario correcta, como para seleccionar obligatoriamente tipos de operación, ajustar rangos de precio automáticamente según el tipo de operación, o para mantener valores dentro de los límites establecidos en los sliders.

3.4.- Navegación clara entre las pantallas y gestión de estados (provider)

El estado de la aplicación se gestiona mediante clases de modelo, principalmente la clase Filtros, que encapsula todos los criterios de búsqueda.

Los filtros se pasan entre pantallas utilizando la navegación con go_router, evitando duplicaciones de estado.

Esta estructura permite una futura integración sencilla de herramientas de gestión de estado como Provider si la aplicación creciera en complejidad.

La navegación entre las distintas pantallas se realiza mediante la librería go_router.

Esto permite una navegación clara y controlada entre: pantalla principal, filtros avanzados y resultados de búsqueda. Además, se mantiene el contexto de los filtros al volver atrás, mejorando la experiencia de usuario.

3.5 – Estructura del proyecto conforme se ha realizado en el resto de proyectos (lib/models, lib/routes, lib/widgets, lib/provider, lib/pages)

El proyecto sigue una estructura modular:

- lib/pages: pantallas principales de la aplicación.
- lib/models: modelos de datos como Filtros y Vivienda.
- lib/widgets: componentes reutilizables.
- lib/routes: configuración de rutas.
- lib/provider:
- lib/dev: seeder para pasar datos a la colección de firestore.

3.6.- Diseño de la interfaz

El diseño de la interfaz se ha realizado priorizando la claridad y la accesibilidad.

Se ha utilizado una paleta de colores suaves asociada a la sostenibilidad, con un predominio del color verde.

Los elementos interactivos están claramente diferenciados y se han evitado sobrecargas visuales, manteniendo un diseño limpio y funcional.

4.- Pruebas

Caso	Descripción del caso	Resultado esperado	Resultado obtenido	Ok/KO	Mensaje de error
Búsqueda	En Home, pulsar “Buscar” sin seleccionar tipo de operación (Comprar/Alquilar).	No navega a Resultados. Se muestra aviso indicando que hay que seleccionar operación.			Selecciona Comprar o Alquilar
Pulsar un filtro rápido	En Home, pulsar filtro rápido “Precio (€)” sin seleccionar operación.	No abre el diálogo de precio. Se muestra aviso indicando que primero hay que seleccionar operación.			Selecciona Comprar o Alquilar primero
Selección de provincia y municipio	Seleccionar Provincia y después Municipio. Cambiar la Provincia a otra distinta.	El Municipio se reinicia a vacío (null) para evitar valores incoherentes con la nueva provincia.			
Activación de filtros de adaptabilidad y transporte público	En Filtros Avanzados, activar “Vivienda adaptada”, marcar “Mayores” y “Transporte público” y pulsar “Aplicar filtros”.	Navega a Resultados y se muestran viviendas que cumplan esos filtros (o mensaje “sin resultados” si no hay).			
Slider de economía	En Filtros Avanzados, cambiar “Tipo de operación” a “Comprar” y revisar el RangeSlider de “Precio”.	El rango de precio se ajusta a los límites de compra (mínimo/máximo de compra) y no permite valores fuera.			
Botones filtros avanzados y cerrar	En Resultados, abrir Filtros Avanzados y pulsar “Cerrar”.	Vuelve a la pantalla anterior (Home o Resultados)			

		según el origen, sin crashear y manteniendo filtros previos si procede.			
--	--	---	--	--	--