

Formulario de SerralInnova

En esta práctica se creará un formulario digital completo para la empresa SERRAINNOVA (inmobiliaria ecosostenible), con el lenguaje de flutter.

ALUMNO
Melisa Hilario

CURSO
11 ene 2026

https://github.com/melisahilario/Proyecto_SerralInnova_FormularioX_HilarioEscribano_Melisa

Índice

1.- Análisis.....	2
1.1.- Descripción del problema y del formulario asignado (objetivo, usuario objetivo y contexto).....	2
1.2.- Requisitos funcionales qué debe cumplir el formulario asignado.....	2
1.3.- Incluye un diagrama de Modelo entidad-relación debe contener:.....	4
1.4.- Diccionario de datos: campos, tipo, obligatoriedad, validación y ejemplo de valor....	5
2.- Diseño UI.....	6
2.1.- Mockups del diseño final de las pantallas.....	6
2.2.- Mapa de pantallas/flujo de navegación (cómo se llega a cada una).....	9
2.3.- Breve guía de estilo: paleta de colores, tipografías, tamaños y estilos de componentes (botones, campos, etc.).....	9
3.- Desarrollo.....	10
3.1.- Desarrollar el formulario en base al análisis realizado.....	10
3.2.- Debe tener persistencia los datos registrados en el formulario.....	10
3.3.- Validación de campos obligatorios, formatos, rangos. Mostrar errores de forma clara junto al campo y evitar registrar información errónea o inválida.....	10
3.4.- Navegación clara entre las pantallas y gestión de estados (provider).....	10
3.5 – Estructura del proyecto conforme se ha realizado en el resto de proyectos (lib/models, lib/routes, lib/widgets, lib/provider, lib/pages).....	11
3.6.- Diseñar preferentemente móvil o web.....	11
4.- Pruebas.....	11
4.1.- Se debe en este punto un plan de pruebas manual con 6 casos como mínimo de funcionalidad del formulario (validaciones, obligatoriedad de los campos, ...).	11

1.- Análisis

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

La inmobiliaria SERRA INNOVA necesita un canal digital claro, accesible y eficiente para recibir y gestionar consultas de usuarios sobre viviendas sostenibles, alquiler social y trámites relacionados, de forma que las solicitudes se clasifiquen, se atiendan con rapidez y se respete su enfoque social y ecosostenible.

Crearemos un formulario para permitir a cualquier persona enviar una consulta y clasificarla, el cliente recibirá confirmación automática con número de ticket y se generará interna mediante la creación de un ticket.

Los usuarios que accederán al formulario son:

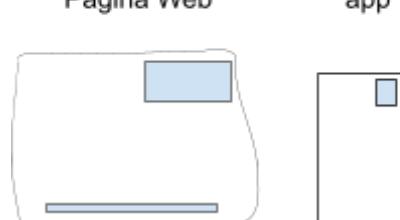
- Personas buscando información sobre viviendas
- Propietarios que quieran publicar una vivienda
- Personas que necesiten realizar trámites administrativos o reportar incidencias técnicas.
- Personas con dudas.

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

Se crea una Página web y Aplicación para atender al cliente para formar una mejor comunicación.

Opciones de entrada al formulario

Para hacer al formulario tendrás las opciones desde un botón con un texto para hacer entender que esta es la opción del formulario o desde un pie de página en el caso de portátiles o pantallas grandes, para la aplicación móvil usaremos un ícono que haga referencia al formulario para acceder.



Pediremos al usuario que elija entre un grupo de consultas, para hacer un filtrado. Antes de empezar a llenar el formulario. Opciones del tipo de consultas:

- Información sobre viviendas sostenibles disponibles.
- Alquiler social y acceso a la vivienda para colectivos vulnerables.
- Publicación de una vivienda por parte de propietarios comprometidos con la sostenibilidad.
- Documentación y trámites administrativos.
- Incidencias técnicas relacionadas con la web o la aplicación.
- otras consultas generales.

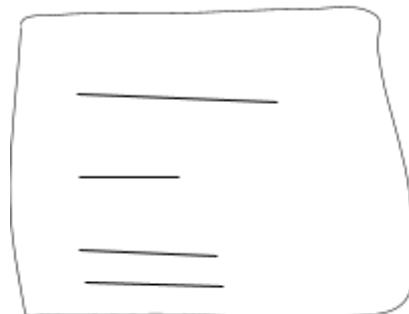
Panel para filtrar la consulta



Después de saber el motivo seleccionado de su consulta pediremos al usuario se obtendrán los datos mediante un formulario con los datos mínimos y tendrá un diseño claro, con lenguaje sencillo:

- Nombre y apellidos.
- Correo electrónico.
- Teléfono de contacto (opcional).
- Tipo de consulta (previamente seleccionado).
- Descripción breve de la consulta o solicitud.
- Aceptación de la política de protección de datos.

Pantalla del formulario

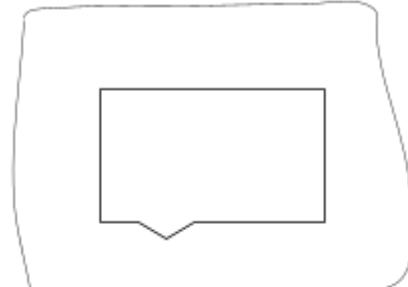


Se necesita que los campos que se rellenen en el formulario siempre cumplan con el criterio, es decir que en el nombre no puede haber números o que en la opción del teléfono solo pueden usar números. Recordar al cliente que marque la opción de Aceptar la política de protección de datos.

Después de que el usuario rellene los datos se generará una respuesta para confirmar al cliente que su consulta será atendida, siendo amigable con el usuario. De esta consulta se genera un ticket con un número de referencia del cliente con la fecha del día que se creó y el motivo del mismo para uso interno.

- Confirmación de que la consulta ha sido recibida correctamente.
- Asignación de un número de referencia o ticket de seguimiento.
- Información sobre el tiempo estimado de respuesta.
- Mensaje de agradecimiento alineado con los valores sociales y sostenibles de SERRA INNOVA.

Panel de confirmación

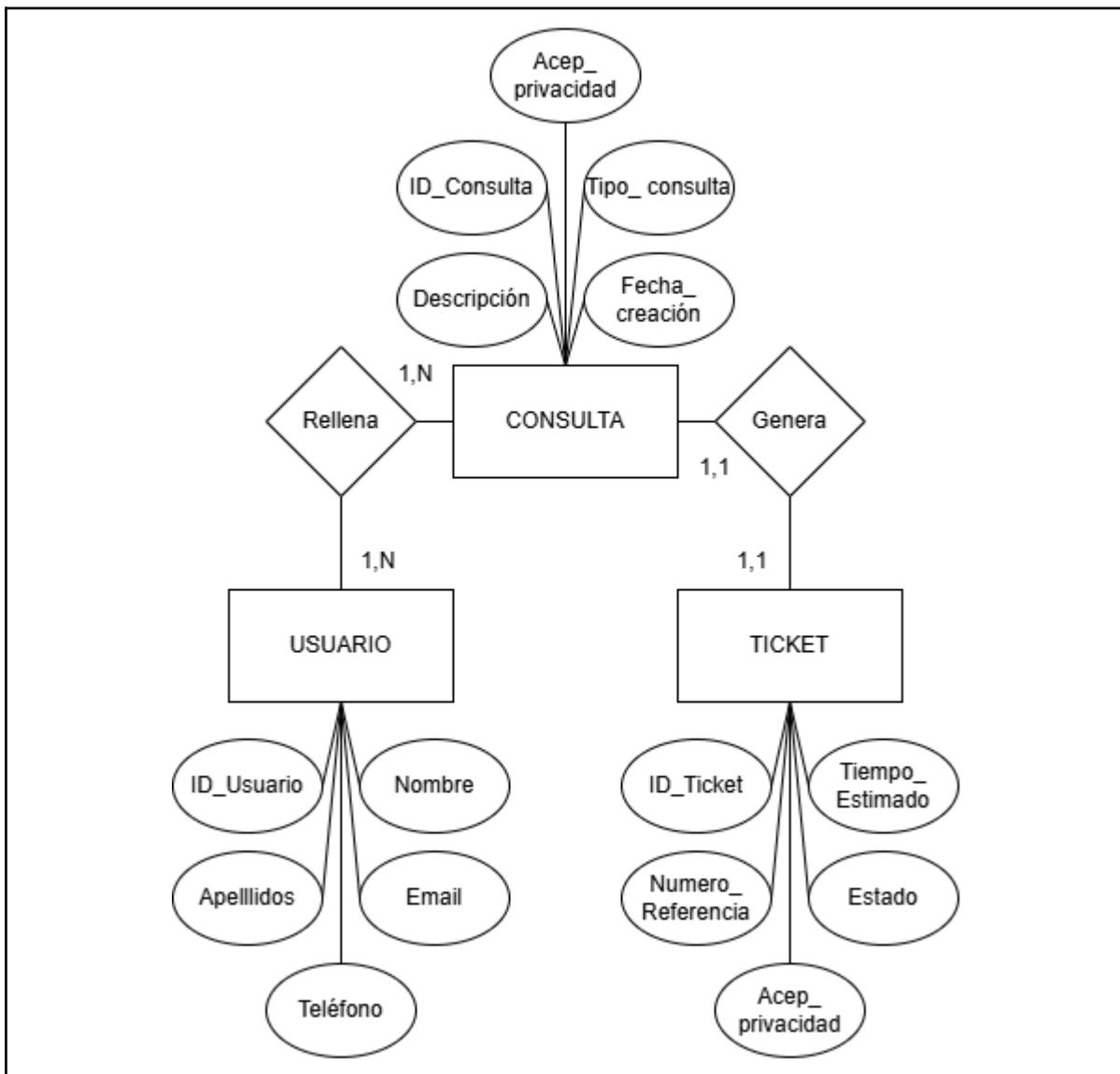


1.3.- Incluye un diagrama de Modelo entidad-relación debe contener:

- Entidades con atributos principales.

Formulario -

- Claves
- Cardinalidades



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

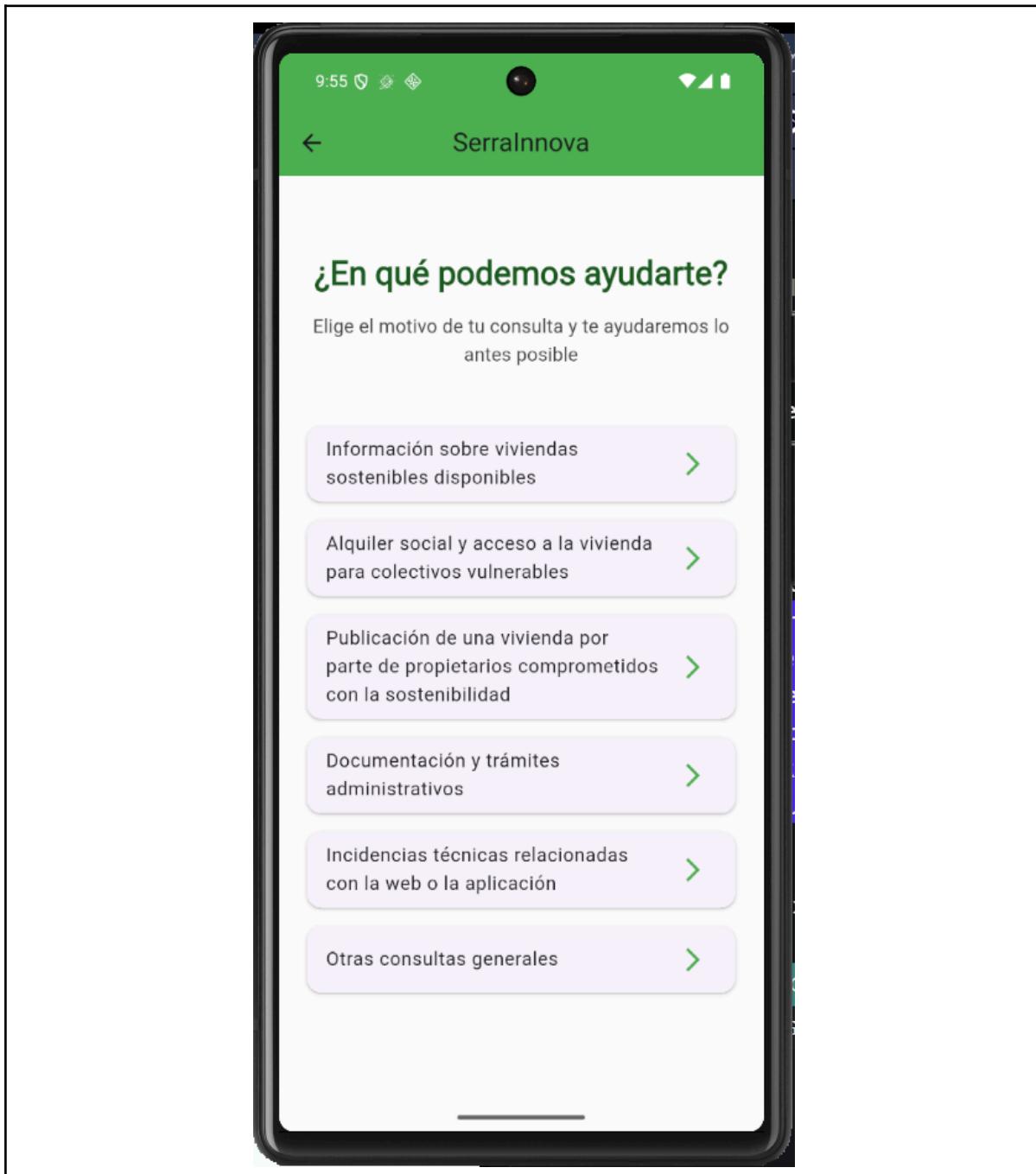
Datos de Formulario:

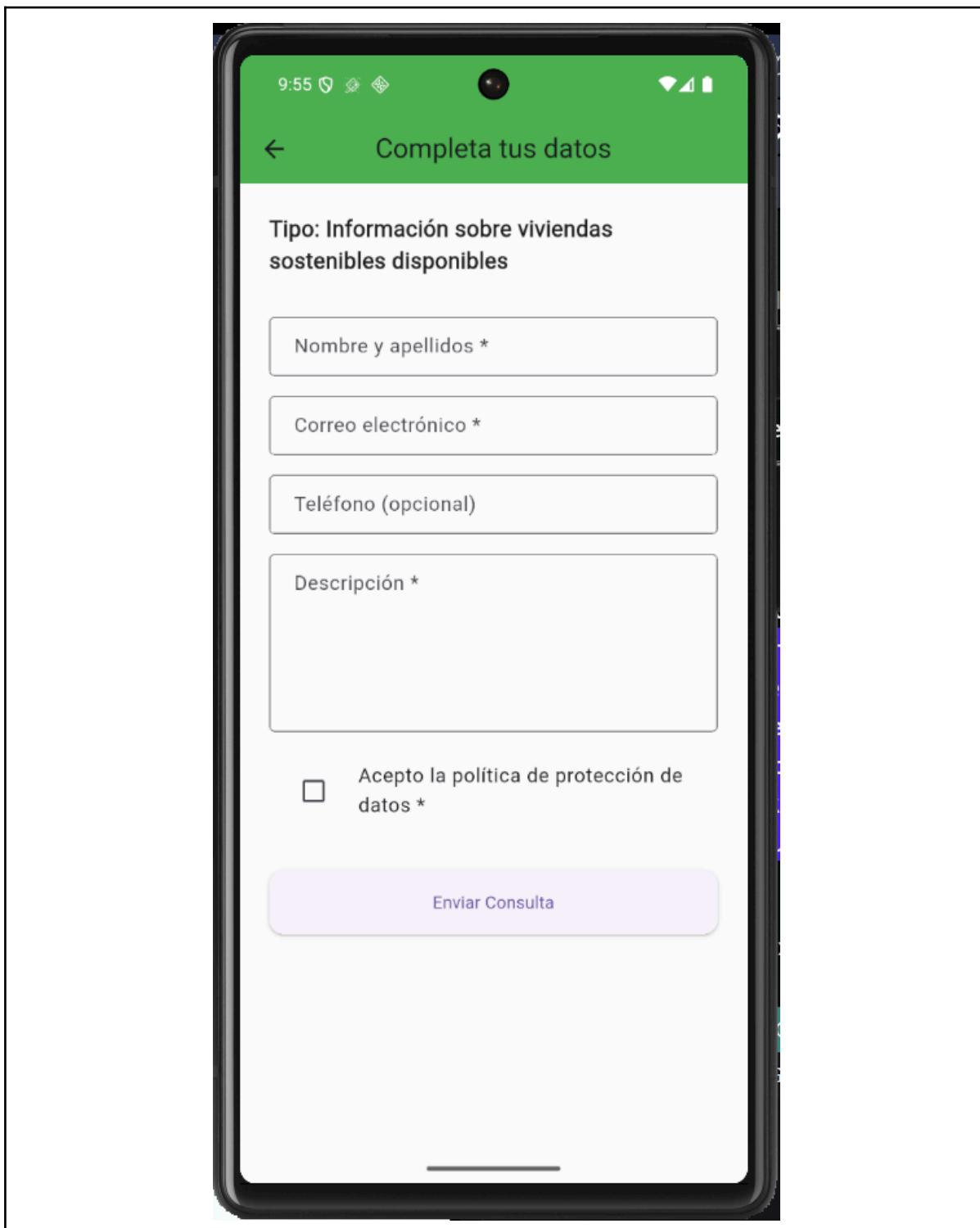
Campo	Tipo	Obligatorio	Validación	Ejemplo	Mensaje de error
Nombre y Apellidos	String	Sí	Solo letras y espacios, longitud. No números ni símbolos especiales.	Juan Pérez García	"El nombre solo puede contener letras y espacios."
Correo Electrónico	String	Sí	Formato de email válido	juan.perez@example.com	"Formato de correo electrónico inválido."
Teléfono de Contacto	String	No	Solo números	+00 000000000	"El teléfono solo puede contener números y debe tener al menos 9 dígitos."
Tipo de Consulta	String	Sí	Debe seleccionarse de una lista predefinida	Alquiler social y acceso a la vivienda para colectivos vulnerables	"Debe seleccionar un tipo de consulta válida."
Descripción Breve	Texto	Sí	Longitud, caracteres concretos	Un texto explicando la consulta	"La descripción no puede estar vacía."
Aceptación de la Política de Protección de Datos	Boolean	Sí	Debe marcarse como verdadero (checkbox).	Verdadero (marcado)	"Debe aceptar la política de protección de datos para continuar."
Fecha de Creación	Date	Sí	Generada automáticamente	2026-01-08 09:03:00	
Número de Referencia (Ticket)	String	Sí	Generado automáticamente	TICKET-0001	

2.- Diseño UI

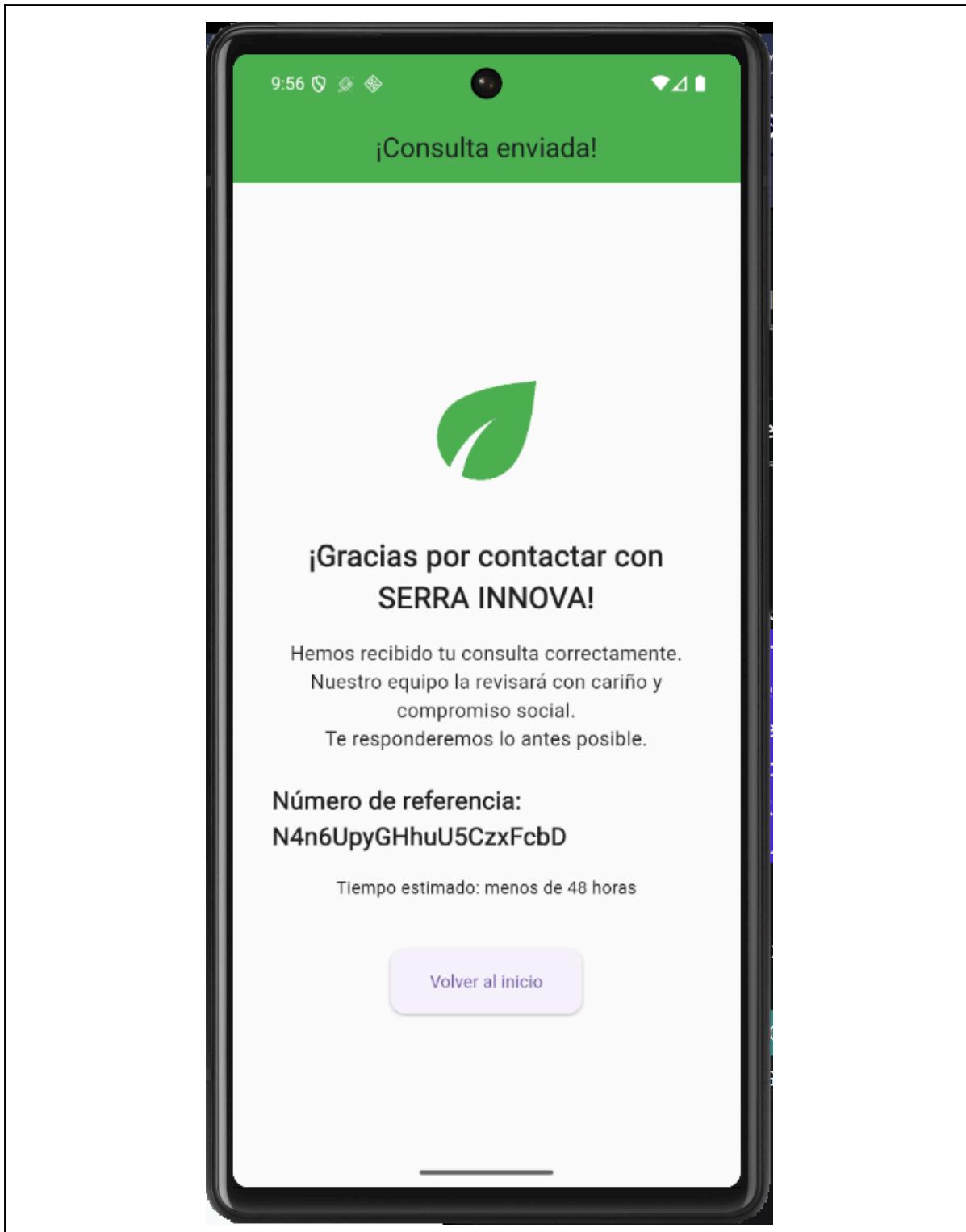
2.1.- Mockups del diseño final de las pantallas.

Pantalla 1: Selección del tipo de consulta (punto de inicio claro y accesible)



Pantalla 2: Formulario de datos mínimos (obligatorios + validaciones)

Pantalla 3: Confirmación automática con ticket y mensaje amable alineado con valores sociales/sostenibles



2.2.- Mapa de pantallas/flujo de navegación (cómo se llega a cada una).

Pantalla Principal → Selección de consulta → Formulario de Consulta → Confirmación → Vuelta a Inicio.

- **Flujo:**

- Desde Inicio: Va a Formulario
- Selecciona el tipo de consulta → Va a Formulario.
- En Formulario: Rellena datos y Pulsar "Guardar" (válido) → Va a Confirmación.
- En Confirmación: Ve mensaje de confirmación y pulsar "Volver" → Vuelta a Inicio.

2.3.- Breve guía de estilo: paleta de colores, tipografías, tamaños y estilos de componentes (botones, campos, etc.).

- **Paleta de Colores** (eco-friendly, inspirado en naturaleza y sostenibilidad):

- Primario: Verde oscuro (#228B22)
- Secundario: Verde claro (#90EE90)
- Fondo: gris claro (#F5F5F5) para legibilidad.
- Error: Rojo suave (#FF6347) Texto: Negro (#000000) para principal, gris (#808080) para secundario.
- Acento: Azul sostenible (#0078D4).

- **Tipografías:**

- General: Google Fonts 'Roboto' (sans-serif, legible). Tamaño base: 16pt.
- Títulos: 'Roboto Bold', 20pt (ej. "Nueva Consulta").
- Labels: 'Roboto Regular', 14pt.
- Textos pequeños: 'Roboto Regular', 12pt (ej. footer).

- **Tamaños y Estilos de Componentes:**

- Botones: alto 48px, ancho full o 120px, radio borde 8px, color primario, texto blanco 16pt bold.
- Campos de Texto: Alto 56px, borde outline (gris), padding 16px, iconosopcionales (ej. phone icon para teléfono). Errores en rojo debajo del campo.
- Dropdown: Alto 56px, mismo estilo que campos, con icono.
- Tarjetas (Card): Margen vertical 8px, elevación 2, radio 8px.
- Iconos: Tamaño 24px, color primario.
- Espaciado: (height: 16) entre elementos para aire.
- Tema general: Colors.green , con soporte dark mode.

3.- Desarrollo

3.1.- Desarrollar el formulario en base al análisis realizado.

El formulario se desarrolla en Flutter. Usa un modelo "Consulta", con tipos de consulta como categorías. El flujo: lista de consultas (home), agregar/editar consulta (formulario), persistencia en Firebase.

3.2.- Debe tener persistencia los datos registrados en el formulario.

Usaremos Firebas. Cada consulta se guarda en la colección 'consultas', con auto-ID como ticket.

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

Las validaciones requieren el nombre, email, descripción, tipo, aceptación. Formato: email con validación, teléfono solo dígitos (9 max). Errores se muestran debajo del campo en rojo.

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

Usa GoRouter para rutas: '/home' , '/tipo-consulta','./formulario' , '/confirmacion' Provider para ContactoService (adaptado a ConsultaService) maneja estado global (lista, cargar/actualizar).

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

- lib
 - models
 - consulta.dart
 - pages
 - pagina_confirmacion.dart
 - pagina_formulario.dart
 - pagina_principal.dart
 - pagina_tipo_consulta.dart
 - provider
 - consulta_provider.dart
 - routes
 - rutas.dart
 - widgets
 - custom_scaffold.dart
 - logo.dart
 - responsive_layout.dart
 - tarjeta_consulta.dart
 - main.dart

3.6.- Diseñar preferentemente móvil o web.

Está diseñado con una preferencia móvil con un diseño vertical con campos de ancho completo, botones grandes y espaciados, y una navegación sencilla.

4.- Pruebas

4.1.- Se debe en este punto un plan de pruebas manual con 6 casos como mínimo de funcionalidad del formulario (validaciones, obligatoriedad de los campos, ...).

Caso	Descripción del caso	Resultado esperado	Resultado obtenido	OK/ KO	Mensaje de error (si tiene el caso)
1	Enviar formulario con todos los campos obligatorios llenos correctamente (nombre, email válido, tipo seleccionado, descripción, aceptación marcada).	Consulta guardada, navega a confirmación con ticket generado.	[A llenar tras prueba]	OK/ KO	No Tiene
2	Intentar enviar sin nombre (obligatorio vacío).	No envía, muestra error junto al campo.	[A llenar]	OK/ KO	"Requerido"
3	Email inválido (ej. "invalid@").	No envía, error en campo email.	[A llenar]	OK/ KO	"Formato de correo inválido"
4	Teléfono con letras (ej. "abc123456").	No envía (input formatter bloquea letras), o error si pasa.	[A llenar]	OK/ KO	"Solo dígitos permitidos"
5	Sin marcar aceptación de política de datos.	No envía, error en checkbox.	[A llenar]	OK/ KO	"Debe aceptar la política"
6	Editar consulta existente, cambiar tipo y guardar.	Actualiza en Firestore, lista en home se refresca.	[A llenar]	OK/ KO	N/A