

Documentación de la aplicación

Doriana Angélica Da Costa Magello
Programación multimedia y dispositivos móviles

Índice

Clase main.....	Página 3
Clase inicio.....	Página 3
Clase menú lateral.....	Página 3
Clase cabecera.....	Página 3
Clase cabecera2.....	Página 4
Clase descripción.....	Página 5
Clase contenido.....	Página 5
Clase iconos.....	Página 6
Clase botón llamar.....	Página 6
Clase Imágenes Verticales.....	Página 7
Clase pie de página.....	Página 7
Clase de cerrar la aplicación.....	Página 8
¿Cómo funciona la aplicación?.....	Página 9
Demostración de la aplicación.....	Página 11
Crear apk.....	Página 15

Clase Main:

La función main es la entrada principal de la aplicación. En este caso, se llama a runApp con una instancia de la clase MyApp. runApp inicia la ejecución de la aplicación Flutter y toma como argumento el widget principal de la aplicación.

Clase MyApp:

Es una clase que hereda de StatelessWidget. Un StatelessWidget es un widget que no tiene estado mutable. En el método build, se devuelve un MaterialApp, que es un contenedor para la aplicación Flutter que sigue las guías de diseño de Material Design.

debugShowCheckedModeBanner: false: Esto desactiva el banner de depuración que aparece en la esquina superior derecha cuando la aplicación se ejecuta en modo de depuración.

Home: Inicio(): Este es el widget principal que se mostrará cuando se inicie la aplicación. Inicio es probablemente otra clase o widget que representa la pantalla de inicio de la aplicación.

Clase Inicio:

En el Scaffold, se configura una barra de aplicación (AppBar) con el título 'Instagram'.

Se agrega un cajón de navegación lateral (drawer) que utiliza la clase MenuLateral.

El cuerpo de la aplicación contiene un widget Center con un mensaje de bienvenida.

Se añaden botones elevados en la parte inferior de la pantalla (persistentFooterButtons), cada uno con un ícono (add y clear).

Se configura un botón de acción flotante (floatingActionButton) con un ícono de persona (person). Al hacer clic en este botón, la aplicación navegará a la pantalla definida por la clase Cabecera.

Clase Menú lateral:

Aporta un drawer personalizado con una imagen, el nombre del autor de la aplicación. Y un AppBar con elementos apropiados.

Posee un menú de navegación lateral que contiene un encabezado de cuenta de usuario con una imagen de fondo y opciones de navegación representadas por elementos de lista. Cada elemento puede tener estilos personalizados y colores de fondo, y uno de ellos está configurado para responder a toques. El menú lateral se integra fácilmente en una aplicación Flutter para proporcionar una interfaz de usuario interactiva y atractiva.

Clase Cabecera:

AppBar Personalizado:

Se crea una barra de aplicación personalizada (AppBar) con un fondo de color específico y sin botón de retroceso (automaticallyImplyLeading: false).

Incluye el nombre de usuario ('nellamonero') con un estilo específico.

Iconos y Acciones:

Muestra iconos como 'restore', 'assessment', 'bookmark_border', 'co_present_outlined', y 'more_horiz'.

Utiliza círculos rojos con números para indicar ciertas notificaciones.

Incorpora un icono de salida (exit_to_app) con un tamaño grande y un manejo de gestos para mostrar un diálogo de confirmación al tocarlo.

Diálogo de Confirmación al Salir:

Al tocar el icono de salida, se muestra un diálogo de confirmación con la opción de aceptar (cerrar la aplicación) o cancelar.

Estructura de la Pantalla:

El cuerpo de la pantalla (body) está compuesto por una columna (Column) que contiene un widget llamado Cabecera2 en la parte superior, seguido por un widget llamado PieDePagina y posiblemente otros elementos en el futuro.

En resumen, la pantalla Cabecera proporciona una interfaz visualmente atractiva con una barra de aplicación personalizada y acciones interactivas, incluyendo un diálogo de confirmación al tocar el icono de salida. Además, utiliza otros widgets secundarios como Cabecera2 y PieDePagina para construir la estructura completa de la pantalla.

Clase Cabecera2:

Estructura Modular y Jerárquica:

La clase está diseñada como un widget Stateless y sigue una estructura jerárquica mediante el uso de un ListView, que organiza los elementos de manera vertical y permite el desplazamiento de contenido.

Imagen de Perfil y Estadísticas:

Destaca una representación elegante de la imagen de perfil del usuario, encerrada en un círculo con un borde distintivo. La inclusión de estadísticas numéricas, como el número de publicaciones, seguidores y seguidos, añade un componente informativo relevante.

Diseño Visual Atractivo:

La disposición visual de la información, como las estadísticas y los botones de acción, se realiza cuidadosamente. Los elementos gráficos, como los círculos y los botones, están estilizados con colores específicos y estilos que contribuyen a la estética general de la interfaz.

Botones de Acción Interactivos:

Se presentan dos botones elevados que invitan a realizar acciones específicas: "Promociones" y "Editar Perfil". Estos botones están diseñados para ser visualmente atractivos y son interactivos, con la capacidad de ejecutar lógica específica al ser presionados.

Interactividad Adicional:

La sección incluye interactividad a través de botones de acción y posiblemente otras lógicas de manejo de eventos, aunque algunas de estas funcionalidades están actualmente comentadas en el código.

Elementos Adicionales Integrados:

Se incorporan elementos secundarios como la descripción del usuario (Descripcion()) y contenido adicional (Contenido()), proporcionando una visión más completa del perfil del usuario.

Clase Descripción:

Estructura y Contenido:

Es un widget Stateless que utiliza un contenedor para organizar la información descriptiva.

Incluye tres textos distintos:

El primero presenta el nombre "Nero" con un estilo de fuente grande y negrita.

El segundo indica "Business" con un tamaño de fuente menor y un color gris para contextualizar la naturaleza de Nero.

El tercero ofrece una descripción más extensa, proporcionando información adicional sobre Nero.

Diseño del Texto y Estilo Visual:

Utiliza un estilo visual coherente, aplicando una fuente específica ('Playpen Sans') a todos los textos para mantener una presentación uniforme.

La descripción se presenta de manera atractiva, considerando tamaños de fuente y negritas para resaltar la información clave.

Espaciado y Margen:

Implementa un pequeño margen alrededor del contenedor que contiene la descripción, lo que contribuye a un espaciado visual agradable y mejora la presentación general en la interfaz de usuario.

Clase Contenido:

Estructura Jerárquica:

La clase está construida como un widget Stateless y sigue una estructura organizada, utilizando una columna para alinear verticalmente varios elementos visuales.

Galería de Imágenes Dinámica:

Se implementa una galería de imágenes dinámica mediante un contenedor con un ListView horizontal. Las imágenes se cargan desde rutas específicas (imagePaths) y cada imagen se asocia con un texto descriptivo (imageTexts). Este diseño permite una presentación visual atractiva y fácil navegación a través de las imágenes.

Botón para Agregar Imágenes:

Se introduce un botón específico para agregar imágenes con un ícono de suma (Icons.add). Al ser presionado, este botón está configurado para activar la lógica de agregar imágenes, aunque esta lógica está actualmente comentada en el código.

Diseño Estético:

Cada imagen se presenta de manera elegante dentro de un círculo con un marco y un fondo blanco, proporcionando una apariencia visual uniforme y agradable.

Lógica para Agregar Imágenes:

El botón para agregar imágenes utiliza un GestureDetector para detectar toques y está preparado para activar una lógica específica al ser tocado. Sin embargo, esta funcionalidad específica se deja a la implementación, ya que la lógica está actualmente comentada.

Texto Descriptivo Asociado:

Se incluye un texto descriptivo asociado a cada imagen, permitiendo al usuario comprender el contenido visual de manera más contextual.

Estilo Visual Consistente:

Se mantiene un estilo visual consistente en toda la sección, lo que contribuye a una experiencia de usuario coherente y estéticamente agradable.

Clase Iconos Fotos:

Estructura Jerárquica:

Es un widget Stateless que compone dos secciones principales en una columna: una fila de iconos y una cuadrícula de imágenes verticales.

Fila de Iconos:

Se presenta una fila horizontal de iconos con un ancho específico (360) y una altura fija (30).

La fila contiene tres iconos con funciones asociadas: 'apps', 'menu', y 'assignment_ind_outlined'.

Cada icono tiene un tamaño de 30 y un color gris (Colors.grey).

Espaciado y Diseño Visual:

Se agrega un espacio vertical de 4 unidades entre la fila de iconos y la cuadrícula de imágenes para mejorar la presentación visual.

Cuadrícula de Imágenes Verticales:

Incluye una instancia del widget Imágenes Verticales que probablemente se encarga de mostrar imágenes organizadas en una cuadrícula vertical.

Clase Imágenes Verticales:

Estructura Jerárquica:

La clase es un widget Stateless que compone una columna que contiene la cuadrícula de imágenes.

Cuadrícula de Imágenes:

Se utiliza un widget GridView.count para construir una cuadrícula de imágenes.

La cuadrícula tiene un total de 9 imágenes dispuestas en 3 columnas (crossAxisCount: 3).

Cada imagen es construida usando el método privado _buildImagen que toma la ruta de la imagen como parámetro.

Tamaño de la Cuadrícula y de las Imágenes:

La altura de la cuadrícula se establece en 300 unidades.

Cada imagen tiene un ancho fijo de 100 unidades.

Diseño Visual:

Cada imagen se presenta dentro de un contenedor con un ancho fijo y usa la propiedad `BoxFit.cover` para asegurar que la imagen cubra completamente el contenedor.

Imágenes Representadas:

Se utilizan diversas imágenes representadas por rutas locales

Clase Botón Llamar:

Estructura Jerárquica:

La clase es un widget `Stateless` que compone una columna que contiene un botón "Llamar" y la instancia del widget `Iconos Fotos`.

Botón "Llamar":

Se presenta un contenedor de 360 x 40 unidades con un borde y un fondo específicos.

El contenido del contenedor es un texto centrado que dice "Llamar" con un estilo de fuente personalizado.

El texto tiene un tamaño de fuente de 16, utiliza la fuente 'DM Sans', es negrita, de color azul y subrayado.

Espaciado y Diseño Visual:

Se agrega un espacio vertical de 2 unidades entre el botón "Llamar" y la instancia de `Iconos Fotos`.

Clase pie de página:

Estructura Jerárquica:

La clase es un widget `Stateless` que compone un contenedor que abarca todo el ancho de la pantalla (`double.infinity`).

Fondo y Distribución de Elementos:

El fondo del contenedor es blanco (`Colors.white`).

Los elementos en el pie de página están distribuidos horizontalmente con un espacio uniforme entre ellos (`MainAxisAlignment.spaceEvenly`).

Iconos y Acciones:

Se incluyen cinco iconos representando diferentes acciones:

Icono de "Inicio" que al ser tocado navega a la página de inicio.

Icono de "Buscar".

Icono para agregar contenido.

Icono de "Favoritos".

Un círculo con un avatar representando al usuario actual.

Estilo Visual del Círculo de Avatar:

Se presenta un círculo con un borde delgado y un avatar en su interior.

El borde del círculo es gris.

El avatar se carga desde una imagen local.

Interactividad:

El icono de "Inicio" utiliza un GestureDetector para detectar toques y navegar a la página de inicio cuando es presionado.

Clase cerrar aplicación:

Estructura Jerárquica:

Utiliza un Scaffold para proporcionar la estructura básica de la pantalla, que incluye una barra de aplicación (AppBar) y un cuerpo centralizado.

Barra de Aplicación (AppBar):

Presenta un título personalizado "Cerrando" en la barra de aplicación.

Define un color de fondo específico (Color.fromARGB(255, 181, 63, 148)) para la barra de aplicación.

Cuerpo de la Pantalla (body):

Centra verticalmente su contenido mediante el uso de Center.

Contiene un conjunto de textos informativos destinados a comunicar al usuario el estado actual de la aplicación durante el proceso de cierre.

Textos Informativos:

"Cerrando la aplicación...": Un mensaje que indica claramente la acción que está teniendo lugar.

Se presenta en un tamaño de fuente de 24 unidades.

Utiliza la fuente personalizada 'DM Sans'.

Está coloreado en azul para destacar y proporcionar un aspecto visual coherente.

"Autor: Dorian Da Costa": Proporciona información adicional sobre el autor de la aplicación.

Se incluye como una forma adicional de comunicación con el usuario.

También utiliza un tamaño de fuente de 24 unidades y tiene un color azul.

Flexibilidad y Adición de Contenido:

Se deja espacio para la posibilidad de agregar más contenido, lo que brinda flexibilidad para futuras personalizaciones o información relevante.

A continuación, se enumeran algunos de los widgets clave que se utilizan:

MaterialApp: Widget principal que configura la aplicación.

Scaffold: Define la estructura de una página típica de la aplicación.

AppBar: Barra de navegación en la parte superior de la pantalla.

Text: Para mostrar texto en diferentes partes de la aplicación.

Elevated Button: Botones elevados que se utilizan para la interacción del usuario.

Icon: Para mostrar íconos en la barra de navegación y otras partes de la interfaz.

AlertDialog: Cuadro de diálogo que muestra un mensaje de confirmación.

Image: Para mostrar imágenes en la interfaz de usuario.

ListView: Widget que muestra una lista de elementos de manera vertical.

GridView: Widget que muestra una cuadrícula de elementos bidimensional

Row: Widget que organiza elementos en una fila.

Column: Widget que organiza elementos en una columna.

Container: Contenedor que permite personalizar la apariencia de los elementos.

Clip Oval: Utilizado para crear imágenes de perfil redondas.

GestureDetector: Para detectar gestos del usuario, como toques.

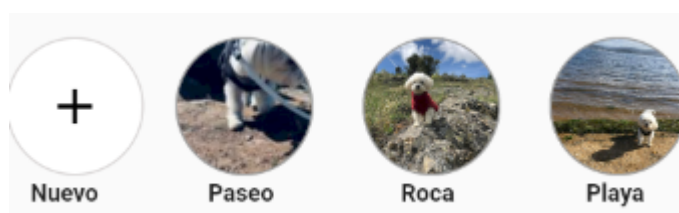
Decoration Image: Utilizado para definir una imagen de fondo para un contenedor.

Center: Widget que coloca su hijo en el centro.

Estos son solo algunos de los widgets utilizados en el código proporcionado. Es posible que haya más widgets anidados dentro de estos, y la cantidad exacta de widgets puede variar dependiendo de cómo se construye la interfaz de usuario y la lógica específica de la aplicación.

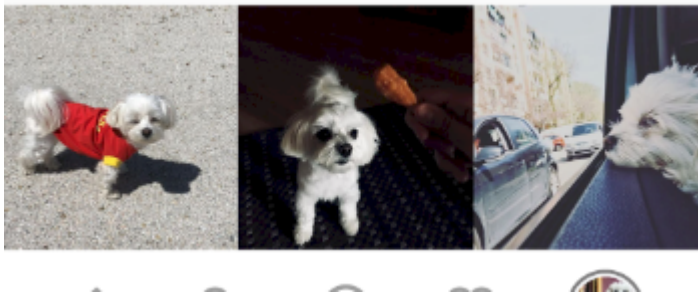
¿Cómo funciona la aplicación?

Puedes visualizar más imágenes, haciendo scroll (deslizando horizontalmente sobre las imágenes). En este apartado son las imágenes agregadas por el usuario en modo historia.



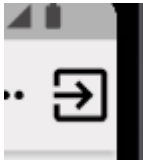
Para visualizar más imágenes de la aplicación, haciendo scroll (deslizando verticalmente sobre las imágenes).

En este apartado son las imágenes agregadas por el usuario a su instagram.

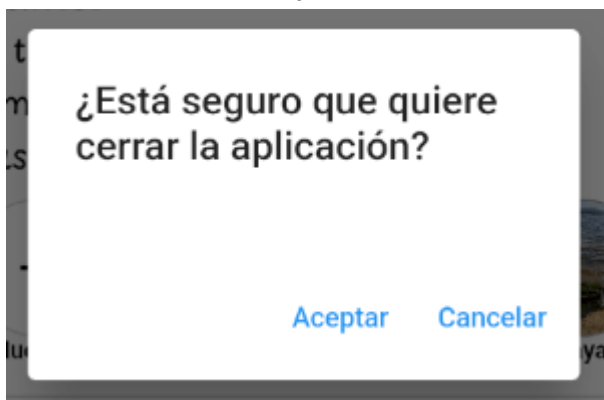


Para salir de la aplicación:

Con el siguiente icono, haciendo click en él, puedes salir de la aplicación.



Se mostrara un mensaje para saber si esta seguro de que se desea salir de la aplicación..



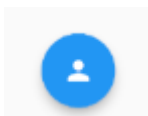
Para volver al menú principal:

Con el siguiente icono, haciendo click en él, vuelves al inicio de la aplicación.



Para acceder a la aplicación de nuestro instagram:

Gracias al siguiente botón podemos hacerle click, y accedemos a la aplicación del instagram

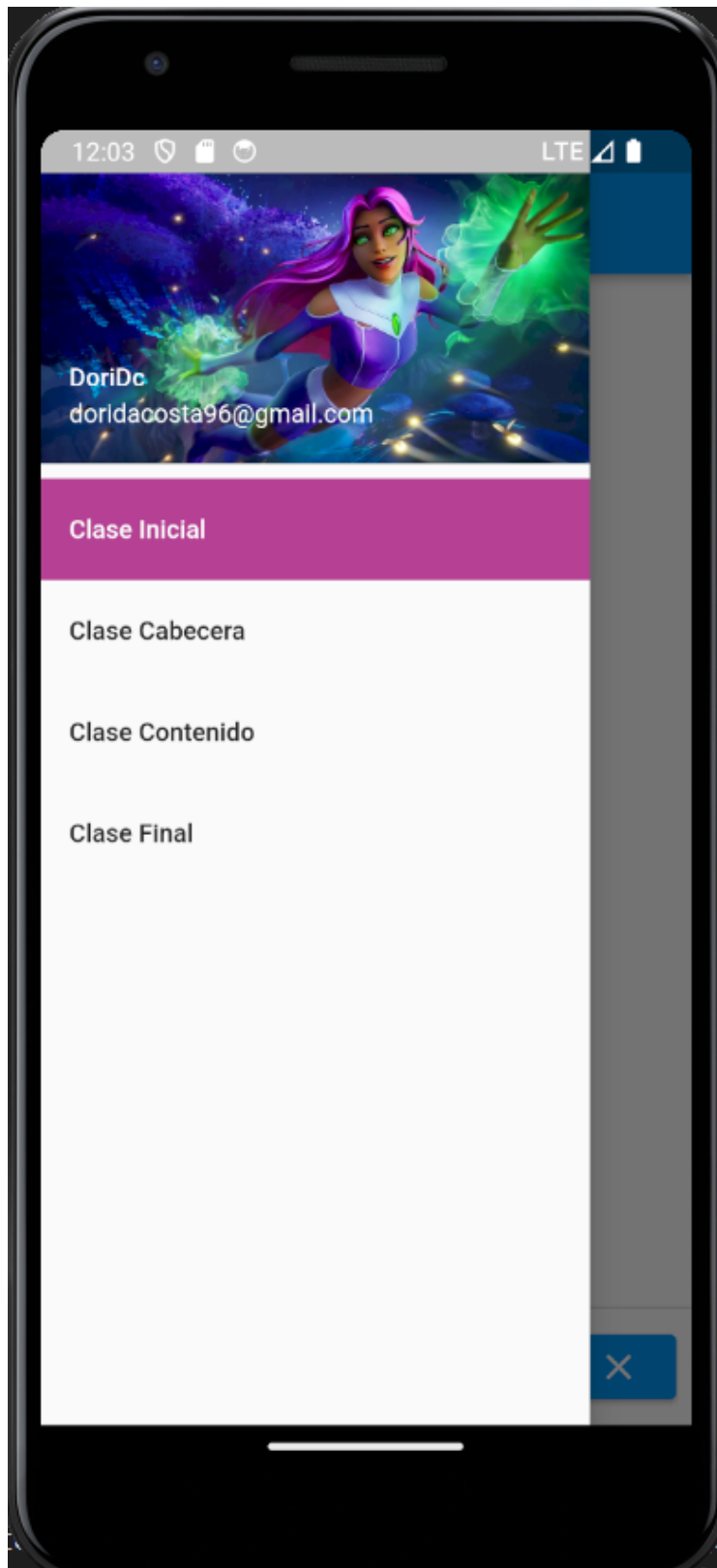


Demostración de la aplicación:

Pantalla Principal:



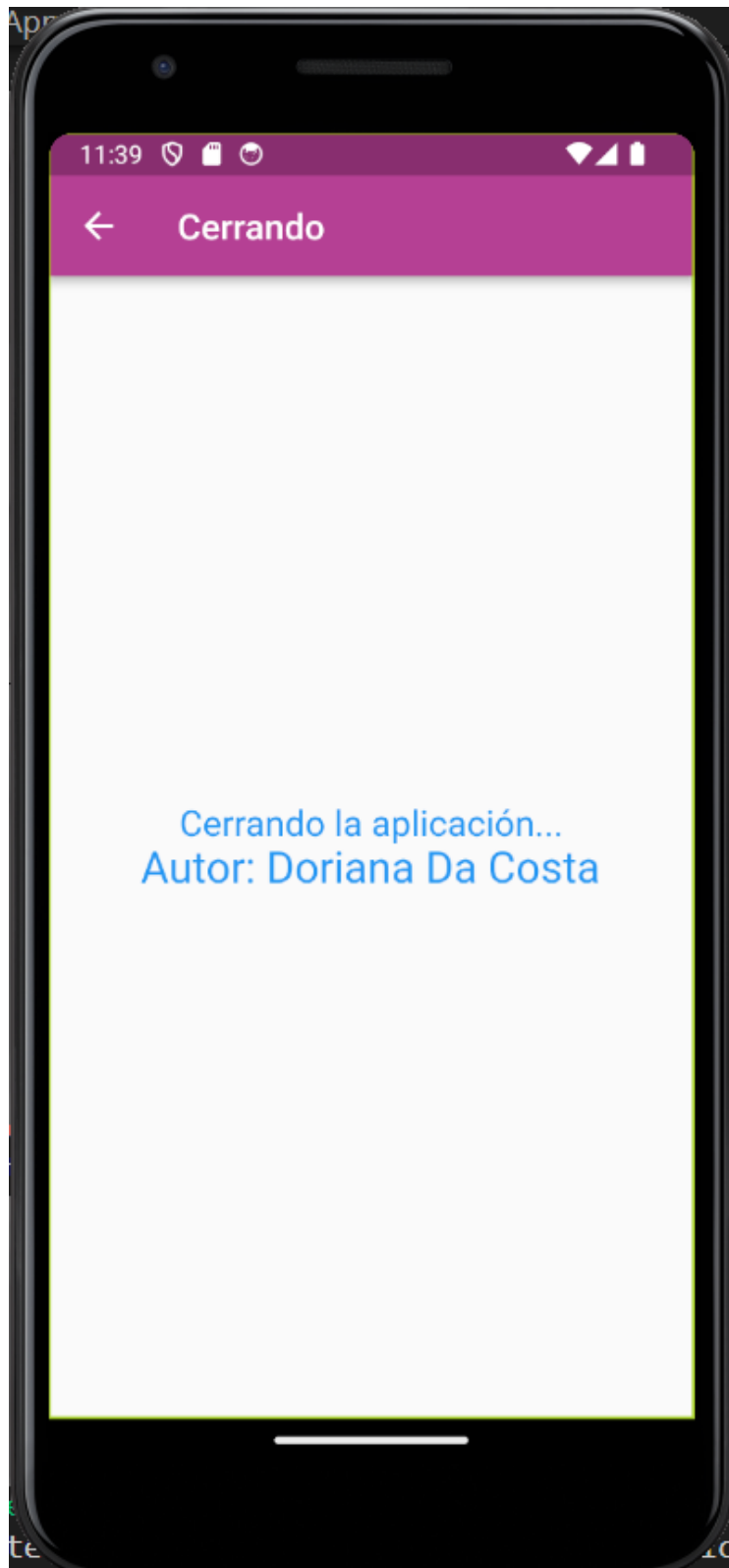
Pantalla Menú lateral:



Pantalla del Instagram:



Pantalla Final:

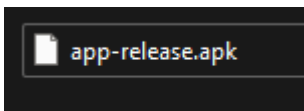


Creamos la apk de la aplicación:

```
PS C:\Users\dorid\OneDrive\Documentos\app_flutter\App-Instagram-Flutter\flutter_application_2> cd flutter_application_2
PS C:\Users\dorid\OneDrive\Documentos\app_flutter\App-Instagram-Flutter\flutter_application_2\flutter_application_2> flutter
build apk
```

```
Running Gradle task 'assembleRelease'... 72,9s
✓ Built build\app\outputs\flutter-apk\app-release.apk (25.2MB).
```

Fichero apk:



Apk instalada en el emulador. Y abre correctamente.

