



# Manual Tecnico

GRUPO 7



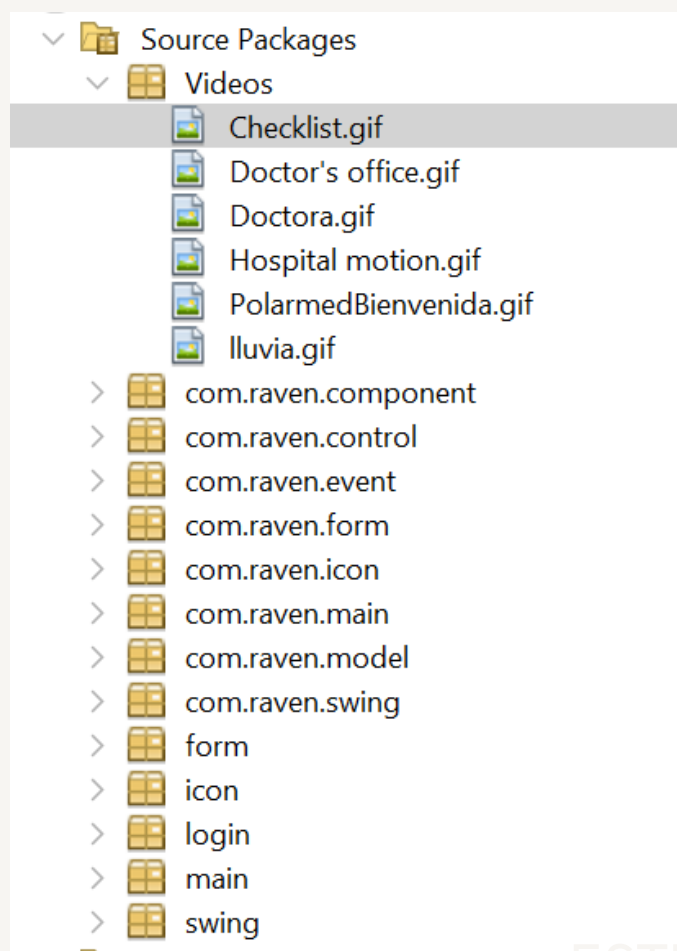


# En este manual se presenta las funcionalidades técnicas del sistema

Tiene como objetivo explicar cómo está  
estructurado el sistema y como este se  
ejecuta

# Videos

Aquí se almacenan las imágenes en movimiento que se proyectan en el panel central del sistema al ejecutarlo



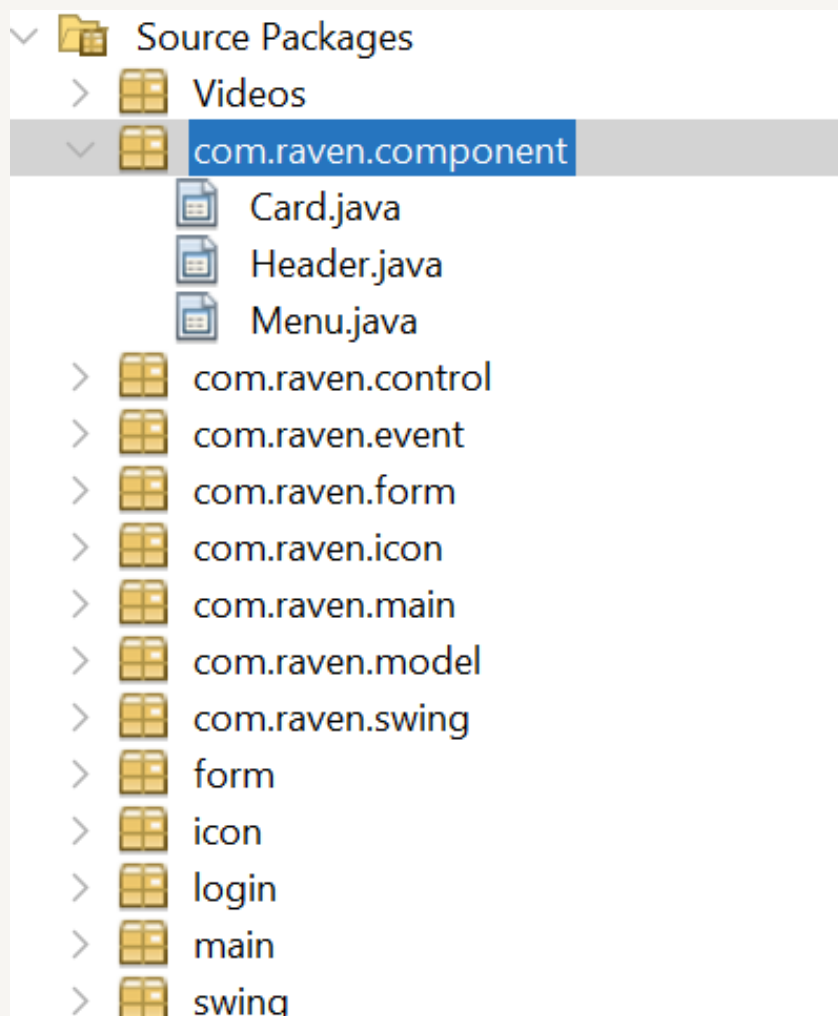
# com.raven.component

Se almacenan Jpanels para su posterior uso en el main.

**Card:** Se define un componente de tarjeta personalizado para una interfaz de usuario de Java Swing, que puede configurarse con datos específicos y muestra un diseño decorativo con un fondo de degradado y elementos gráficos adicionales.

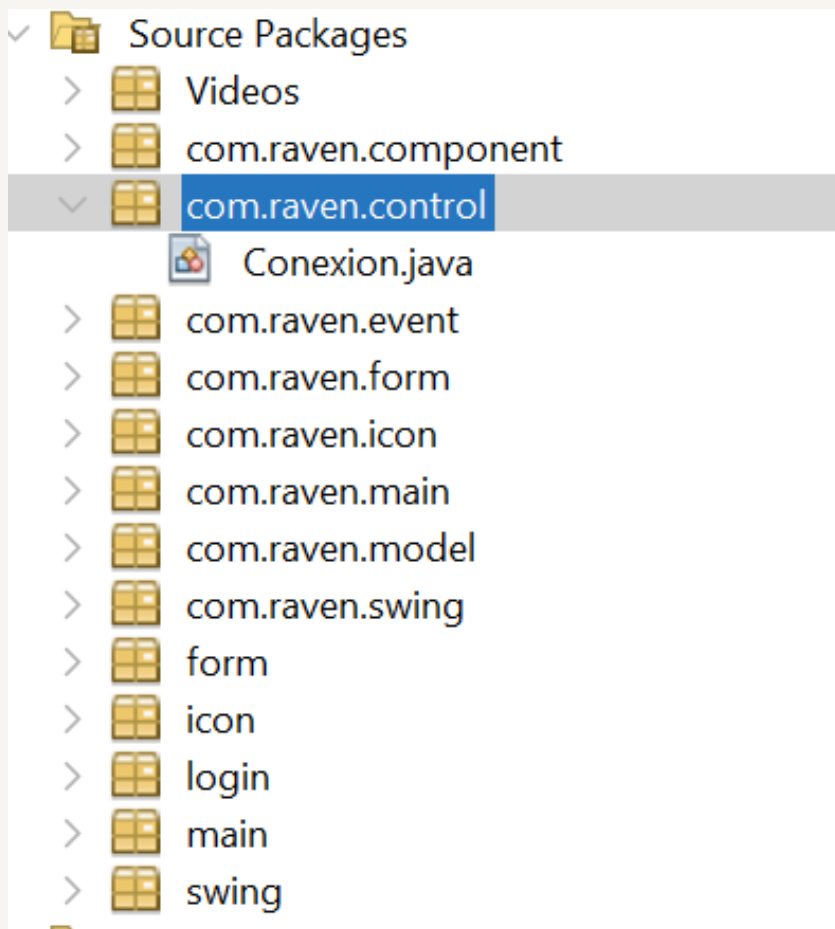
**Header:** Tenía como finalidad ser una barra de búsqueda en la parte superior, pero se optó por no colocarla al final

**Menu:** Se define un componente de menú personalizado para una interfaz de usuario de Java Swing, que puede ser configurado con diferentes elementos de menú y permite arrastrar el marco de la aplicación al hacer clic y arrastrar el menú. Tiene como funcionalidad mostrar las diferentes opciones del programa en el lado izquierdo de la ventana



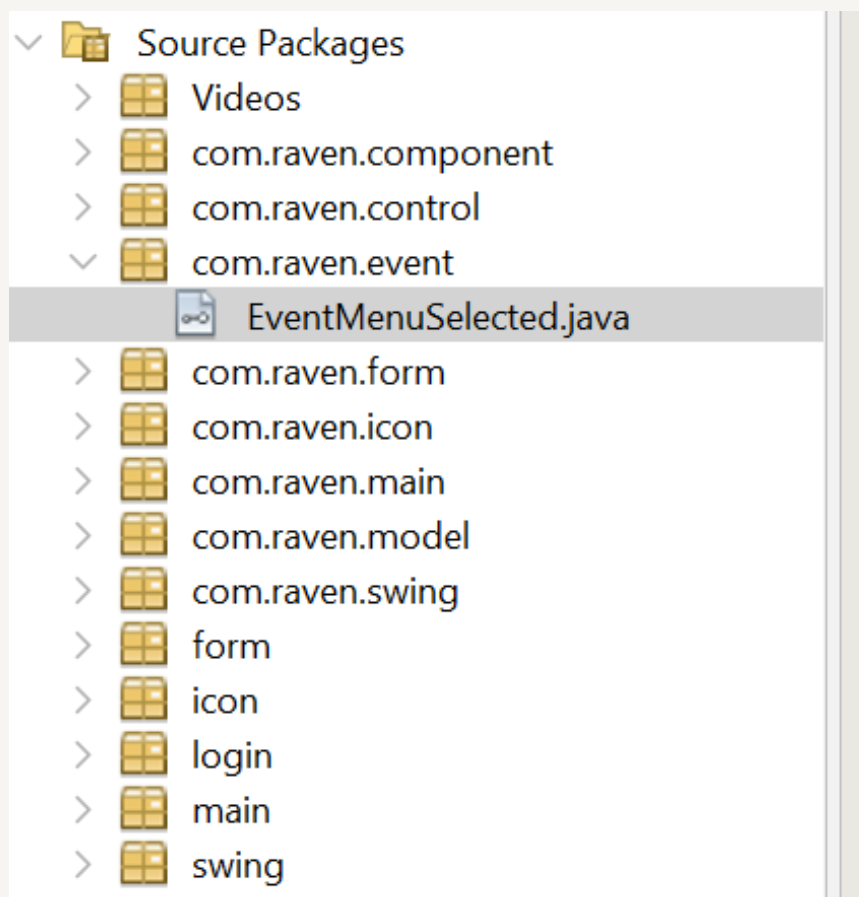
# com.raven.control

Esta clase proporciona métodos para establecer y cerrar la conexión con una base de datos MongoDB utilizando la biblioteca de cliente de MongoDB para Java.



# com.raven.event

Esta interfaz EventMenuSelected define un contrato para los oyentes de eventos de selección de menú, especificando que cualquier clase que implemente esta interfaz debe proporcionar una implementación del método selected, el cual será invocado cuando ocurra un evento de selección de menú, con información sobre el índice del elemento de menú seleccionado.



# com.raven.form

**Form1:** proporciona un formulario interactivo para ingresar los detalles de una compra, guardar esos detalles en una base de datos y enviar un resumen de la compra por correo electrónico en forma de un archivo PDF adjunto.

**Form2:** Forma parte del registro de proveedores y productos que permite guardar los datos en una base de datos MongoDB, generar un PDF con esos datos y enviar ese PDF por correo electrónico.

**Form3:** Forma parte de una aplicación de historial de compras que carga los datos de compra desde una base de datos MongoDB y los muestra en una tabla en la interfaz de usuario.

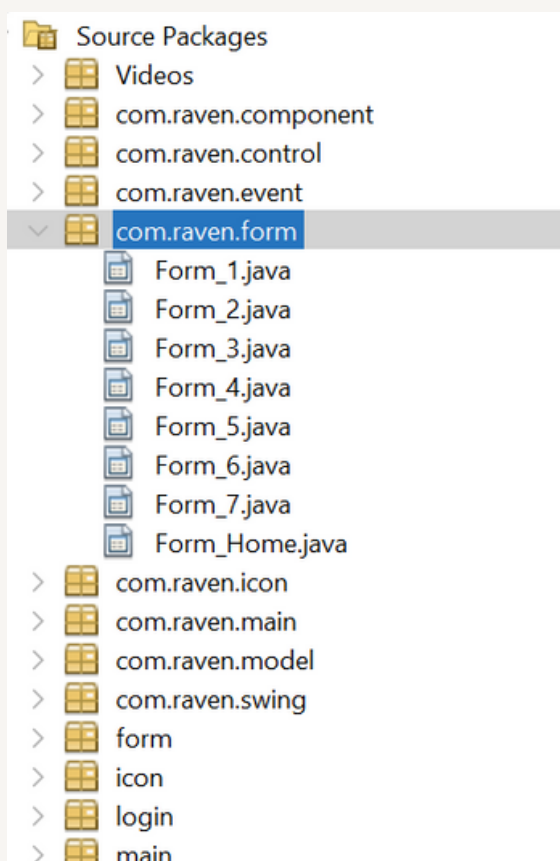
**Form4:** Gestión de proveedores que utiliza una base de datos MongoDB para almacenar y mostrar la lista de proveedores en una interfaz de usuario Swing. Permite al usuario editar y eliminar proveedores, así como actualizar la lista de proveedores mostrada en la interfaz de usuario.

**Form5:** Registro de pacientes que utiliza una base de datos MongoDB para almacenar la información de los pacientes. Permite al usuario ingresar información como el nombre del paciente, el nombre del representante, la dirección y el número de teléfono, y guarda esta información en la base de datos cuando se hace clic en el botón "GUARDAR".

**Form6:** Búsqueda de pacientes que utiliza una base de datos MongoDB para almacenar la información de los pacientes. Permite al usuario buscar pacientes por nombre y ver sus detalles en una ventana de diálogo.

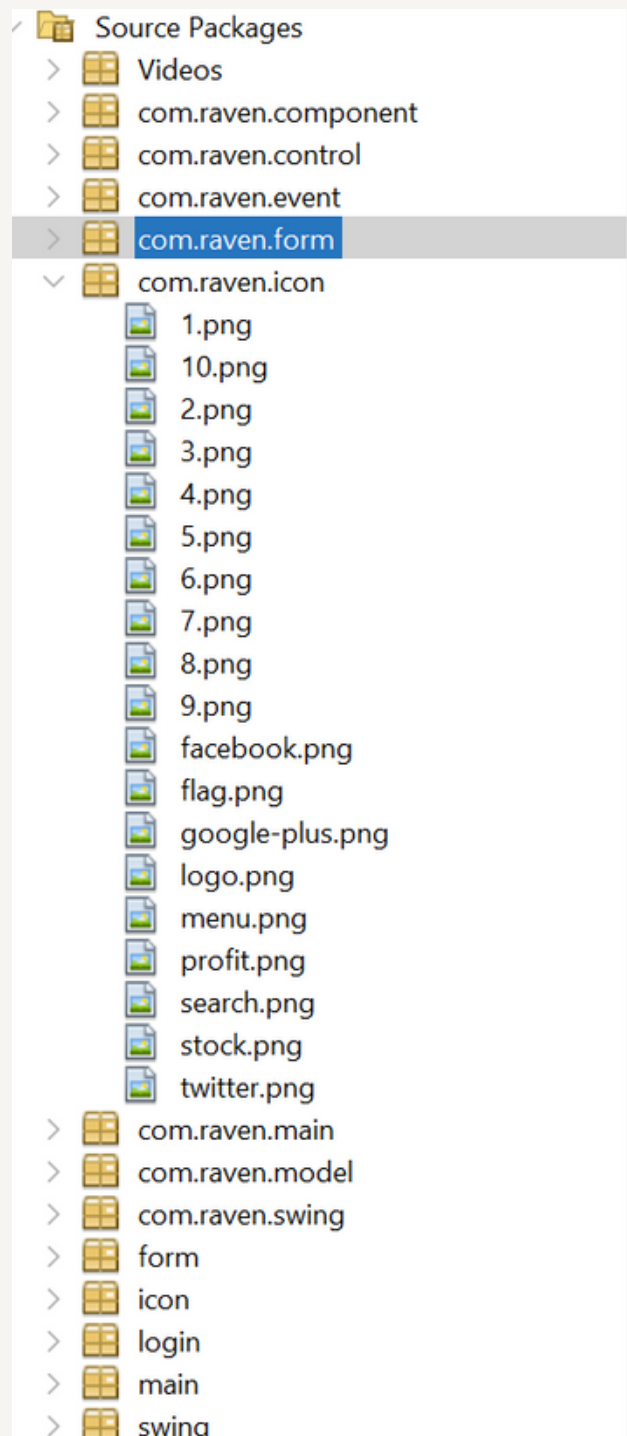
**Form7:** Gestión de pacientes que utiliza una base de datos MongoDB para almacenar y gestionar los datos de los pacientes. Permite al usuario ver una lista de pacientes, editar sus datos, eliminar pacientes existentes y actualizar la lista de pacientes.

**FormHome:** Interfaz de usuario Swing que muestra una serie de gifs en un panel principal. Utiliza un componente Card para mostrar los gifs y un temporizador para cambiar automáticamente los gifs después de un intervalo de tiempo.



# com.raven.icon

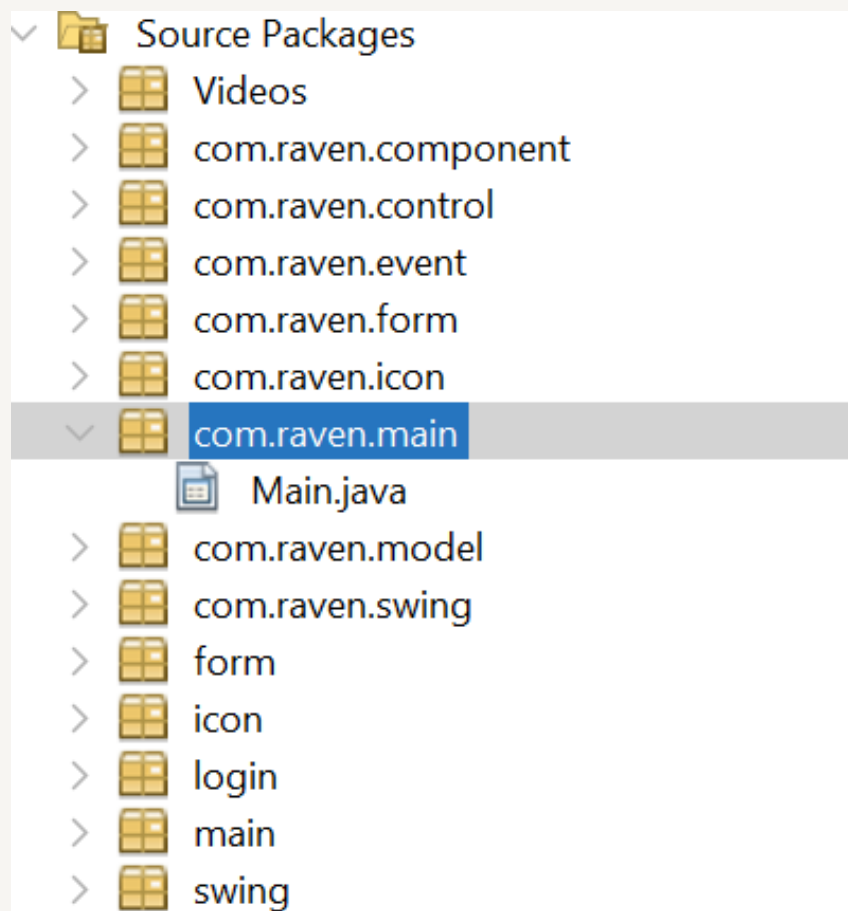
Se guardan todos los iconos usados en el sistema





# com.raven.main

Utiliza un menú lateral para cambiar entre diferentes formas en el panel principal. Cada forma representa una pantalla o funcionalidad diferente de la aplicación.

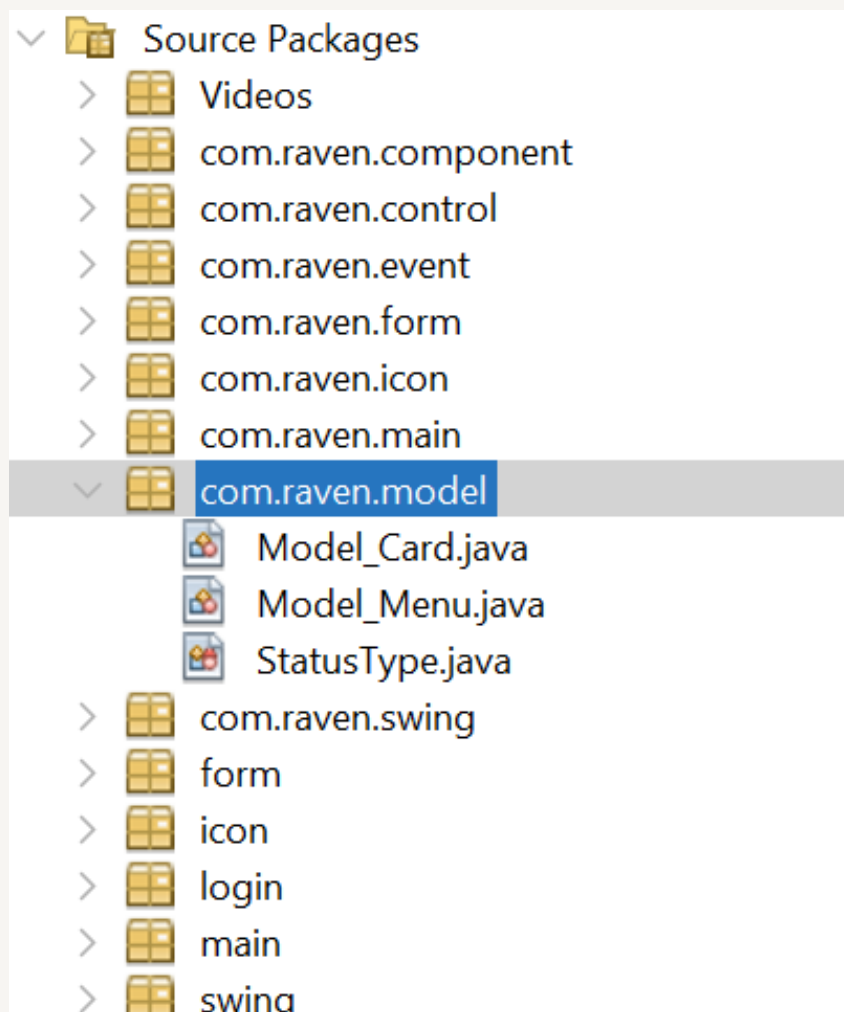


# com.raven.model

**Model\_Card:** Esta clase encapsula los datos relacionados con una tarjeta en una interfaz gráfica, proporcionando métodos para acceder y modificar estos datos. Es común utilizar clases de modelo como esta para representar los datos en aplicaciones de interfaz de usuario, separando claramente la lógica de presentación de la lógica de negocio.

**Model\_Menu:** Esta clase encapsula los datos relacionados con un elemento de menú en una interfaz gráfica, proporcionando métodos para acceder y modificar estos datos, así como para convertir el nombre del ícono en un objeto Icon. La enumeración MenuType proporciona una forma de definir y gestionar los diferentes tipos de elementos de menú de manera consistente.

**StatusType:** Los enumeradores son útiles para representar conjuntos fijos de valores que son conocidos en tiempo de compilación. Proporcionan una forma más segura y legible de trabajar con constantes en comparación con el uso de enteros o cadenas. En este caso, StatusType facilita la comprensión y el mantenimiento del código al definir de manera clara y explícita los posibles estados que pueden tener los objetos en una aplicación.

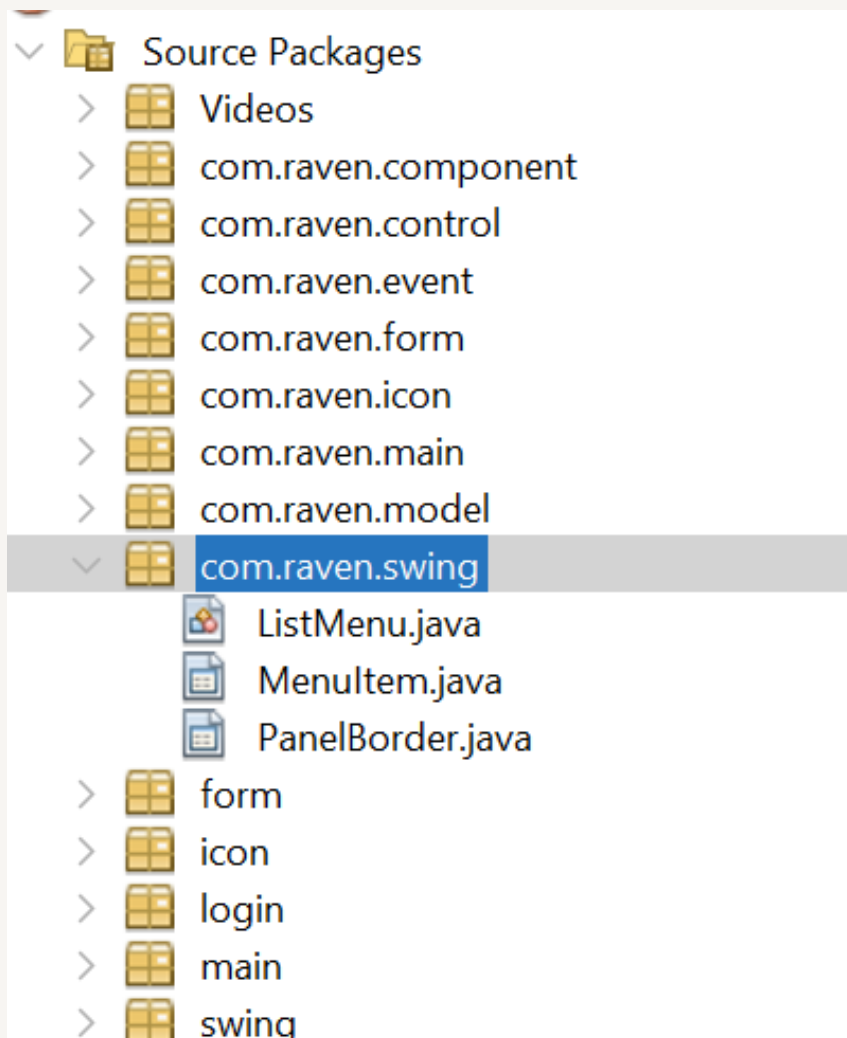


# com.raven.swing

**ListMenu:** Proporciona una lista personalizada que puede contener elementos de tipo Model\_Menu. Permite manejar eventos de clic y movimiento del ratón, y utiliza un renderizador personalizado para mostrar los elementos de la lista.

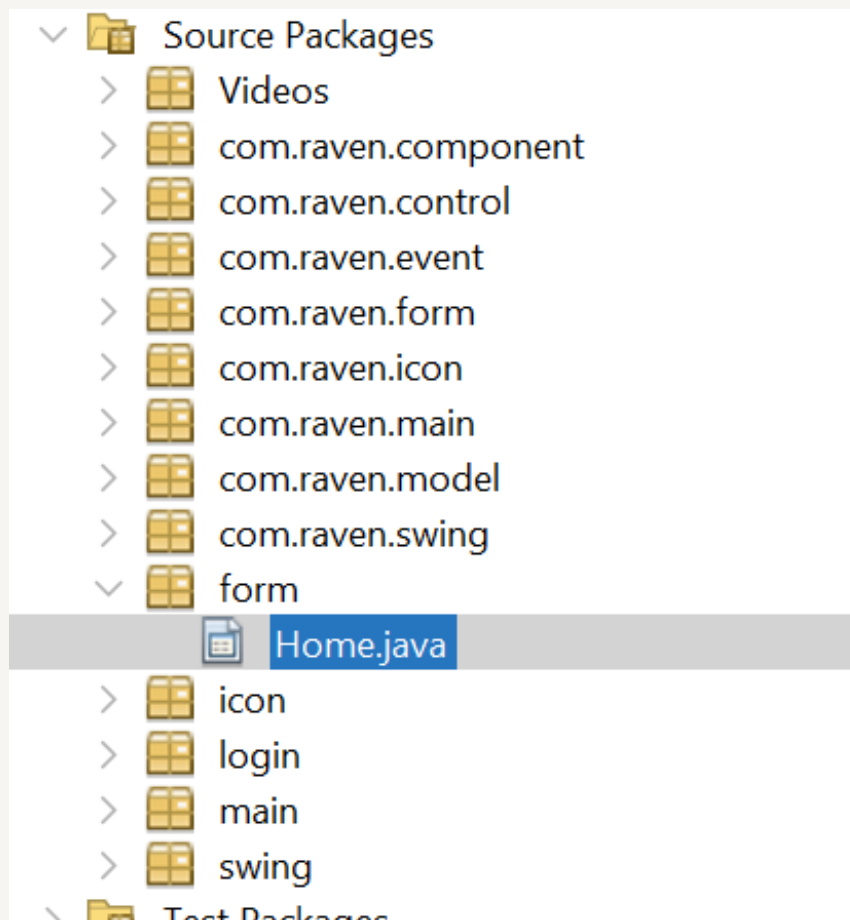
**MenuItem:** Es un componente de menú personalizado que puede representar diferentes tipos de elementos de menú (MENU, TITLE, EMPTY). Ofrece métodos para configurar su estado de selección y manejar eventos de ratón.

**PanelBorder:** Es un componente de panel que tiene bordes redondeados y se utiliza para mejorar la estética visual de la interfaz de usuario.



# form

la clase Home representa la pantalla de inicio de sesión de una aplicación, que incluye una etiqueta de bienvenida y un botón para cerrar sesión. La acción de cerrar sesión está asociada a un evento EventLogin, que se puede establecer mediante el método `setEventLogin()`.



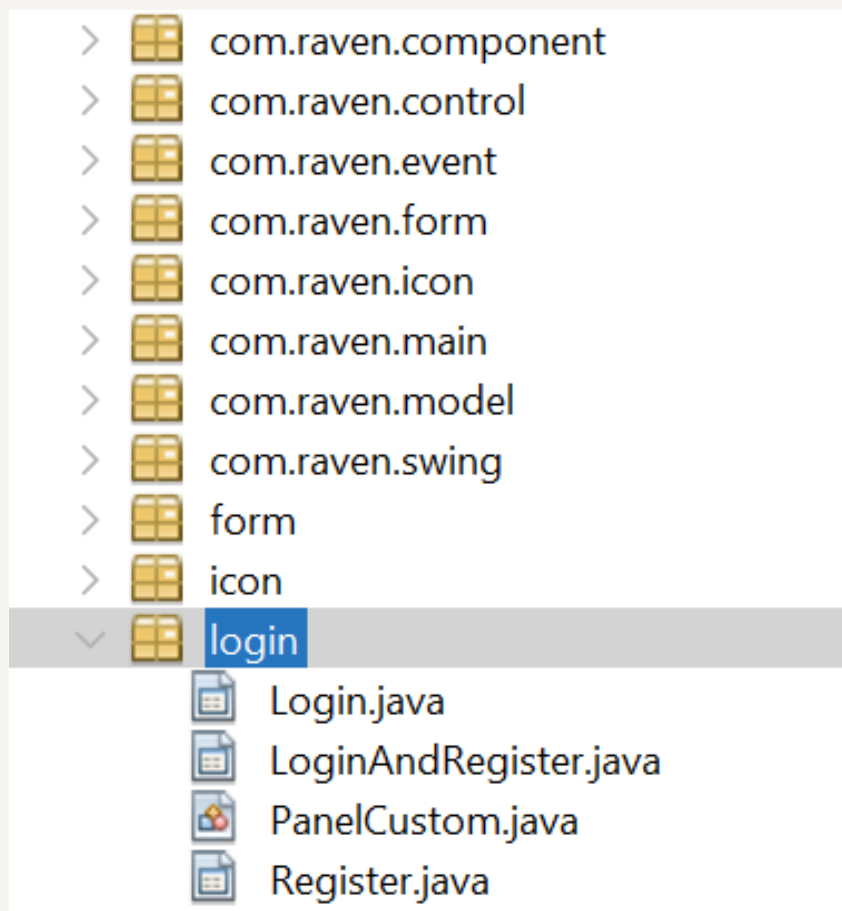
# Login

**Login:** Login representa la pantalla de inicio de sesión de una aplicación. Permite al usuario ingresar su nombre de usuario y contraseña, y luego verificar estas credenciales en una base de datos MongoDB. Dependiendo del resultado de la verificación, se notifica al usuario si el inicio de sesión fue exitoso o no.

**LoginAndRegister:** La clase LoginAndRegister representa una interfaz gráfica que permite al usuario alternar entre los formularios de inicio de sesión y registro mediante animaciones suaves. Además, proporciona la funcionalidad para manejar eventos relacionados con el inicio de sesión.

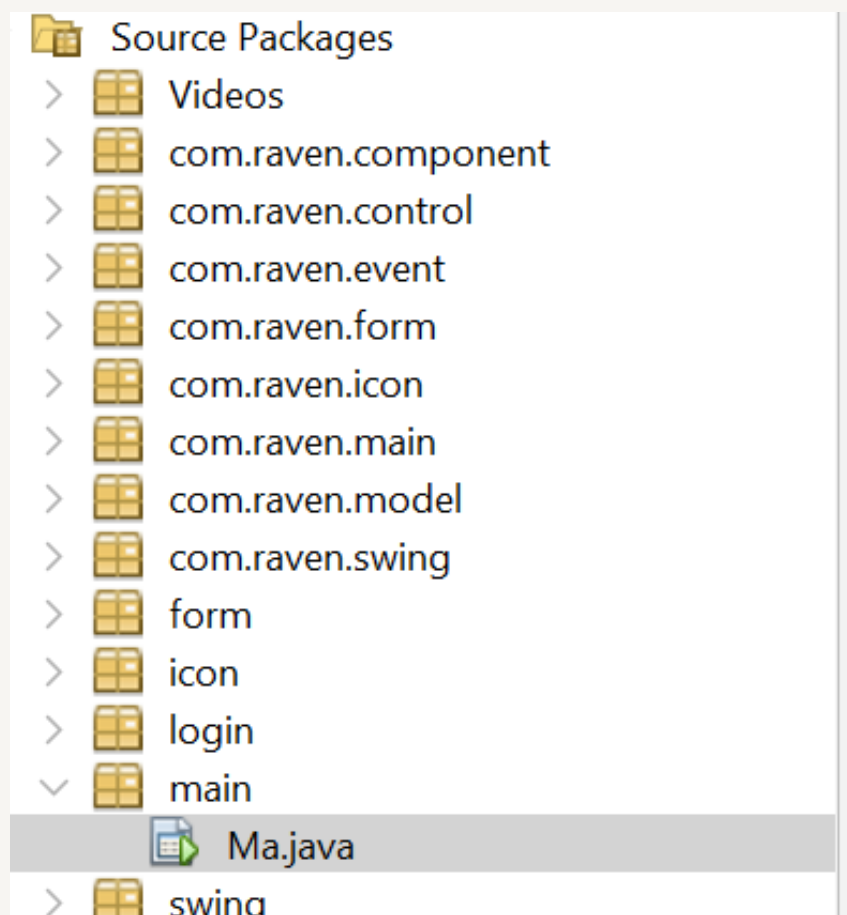
**PanelCustom:** PanelCustom proporciona un panel personalizado que puede tener un nivel de transparencia ajustable. Este panel se utiliza en la interfaz gráfica de usuario (GUI) para crear efectos visuales como paneles semitransparentes o con efectos de desenfoque.

**Register:** Register encapsula la lógica relacionada con el registro de usuarios en la base de datos MongoDB, así como la presentación de la interfaz de usuario correspondiente.



# main

Ma encapsula la lógica relacionada con la interfaz de usuario de la ventana principal de la aplicación, incluido el manejo de eventos de inicio de sesión y cierre de sesión.



# swing

**Button:** Button proporciona un botón personalizado con un borde vacío, sin relleno en el área de contenido y un cursor de mano. Además, el botón tiene bordes redondeados para una apariencia más agradable.

**EventLogin:** la interfaz EventLogin proporciona una forma de definir un contrato para gestionar eventos relacionados con el inicio de sesión y cierre de sesión en una aplicación. Cualquier clase que implemente esta interfaz debe proporcionar la implementación de los métodos loginDone() y logout(). Esto permite una gestión uniforme de estos eventos en diferentes partes de la aplicación.

**Password:** Password personaliza un campo de contraseña en Swing para mostrar un texto de sugerencia cuando está vacío y proporciona una apariencia visual mejorada con bordes redondeados. Esto puede ser útil para mejorar la usabilidad y la estética de las interfaces de usuario que requieren campos de contraseña.

**TextField:** TextField personaliza un campo de texto en Swing para mostrar un texto de sugerencia cuando está vacío y proporciona una apariencia visual mejorada con bordes redondeados. Esto puede ser útil para mejorar la usabilidad y la estética de las interfaces de usuario que requieren campos de texto.

