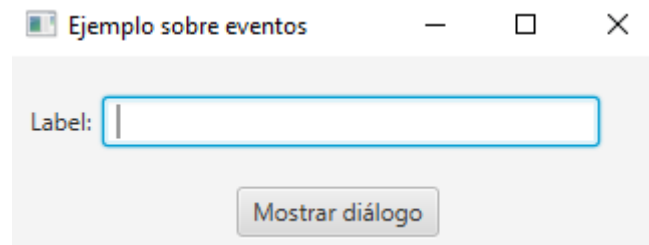


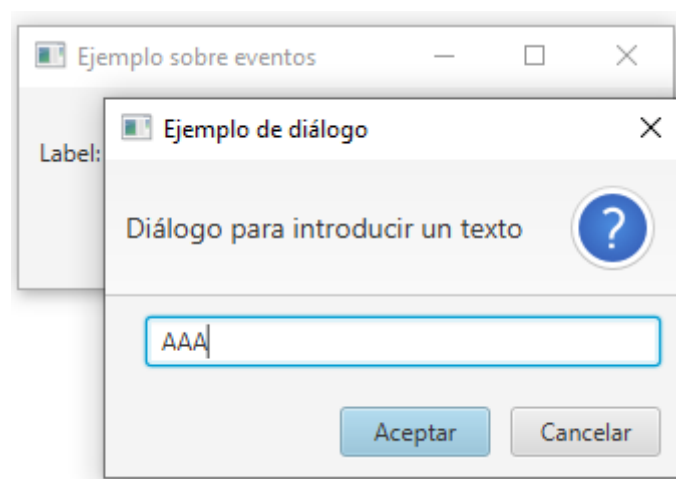
Los ejercicios de este documento están orientados al aprendizaje de diálogos y eventos en JavaFX. Para su realización se creará un proyecto denominado **EjerciciosEventos_Dialogos_Unidad10**

Ejercicio 1

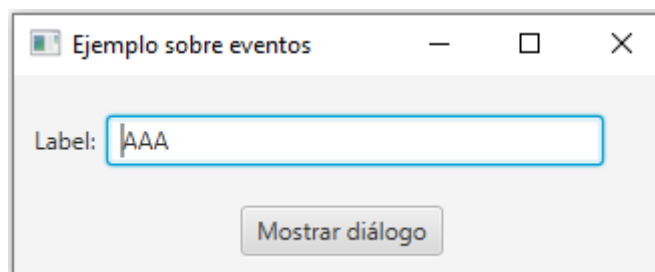
Crea un paquete denominado **eventosValidar** en el cual se incluya un diseño similar al siguiente.



Cuando se hace click en “Mostrar diálogo” se abre la siguiente ventana para introducir un cuadro de texto



El texto del cuadro se almacenará en el campo de texto de la ventana principal cuando el usuario haga click en “Aceptar”.



Ejercicio 2

Añade al ejercicio anterior en el paquete **eventosValidar** las siguientes funcionalidades:

- En el texto del diálogo solamente se pueden añadir letras en mayúsculas. El resto de caracteres no son válidos y cuando el usuario los teclee no se mostrarán.
- Normalmente, la tecla ENTER es un acceso directo al botón “Aceptar” del diálogo y “ESC” del botón cancelar. Impide que el usuario pueda realizar estas acciones y que el diálogo solo se pueda cerrar al hacer click con el ratón en los botones correspondientes.
- Cuando la aplicación detecta el evento que cierra la ventana del diálogo, el foco se situará sobre el campo de texto de la ventana principal.
- Al posicionar el ratón por encima del campo de texto de la ventana principal se cambiará el cursor al tipo “pointer” (un icono con una mano). Esto debe implementarse a partir de un evento y en lugar del selector :hover de CSS.

Ejercicio 3

Crea un paquete denominado **draggableFiles** en el cual se incluya un diseño similar al siguiente.



Debes realizar las dos siguientes operaciones.

- Cuando “Source String” se arrastra a “Destination”, se debe copiar el texto origen en el destino e indicar en el origen que la operación ha terminado, como se muestra debajo. Es la única operación permitida para arrastrar texto y siempre en esta dirección.

Operación drag terminada

Source String



- Es posible arrastrar imágenes del explorador al recuadro en el que se indica esta posibilidad. Este vídeo podría servirte de ayuda: https://www.youtube.com/watch?v=f7KGXUrAH0g&feature=emb_logo. Basta con gestionar los eventos **DragOver** y **DragDropped** y detectar que el tipo de datos que se ha arrastrado es un fichero. La imagen se mostrará en el recuadro creando dinámicamente un **ImageView**.

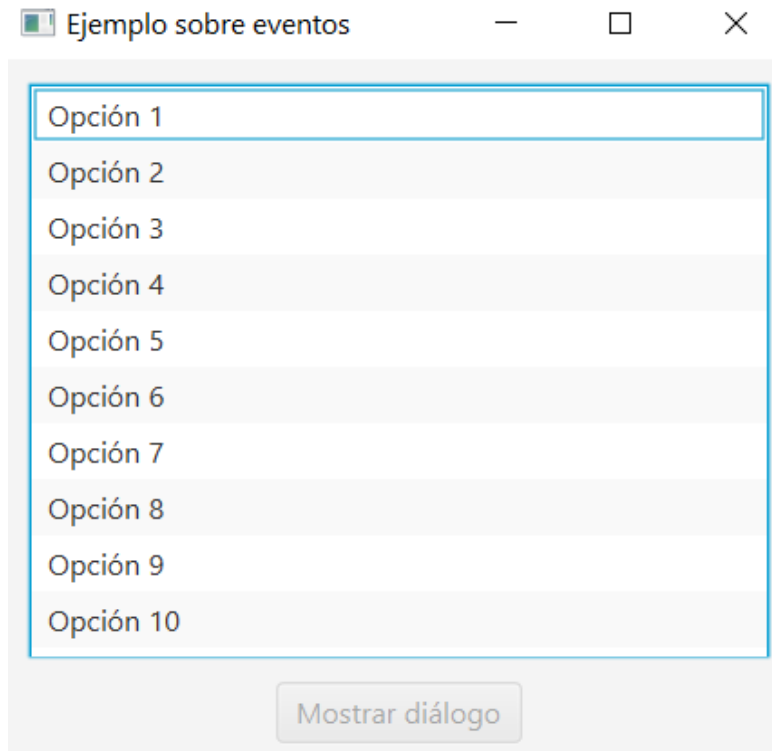
Source String

Destination



Ejercicio 4

Crea un paquete denominado **choiceEjercicio** y realiza un diseño similar al que se muestra a continuación.

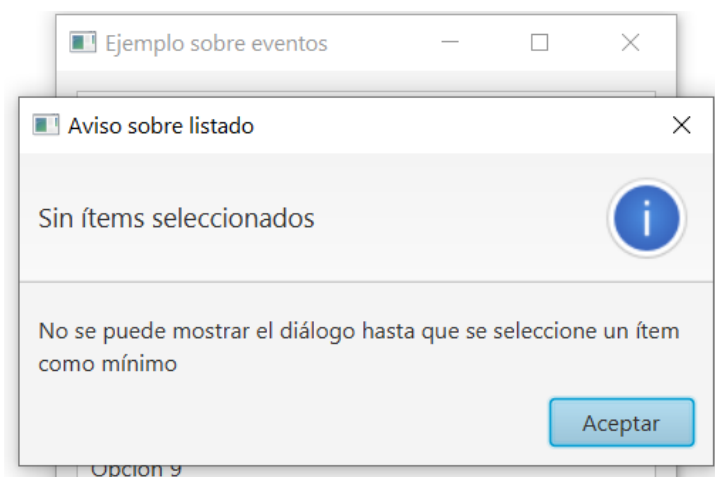


La situación inicial será la siguiente:

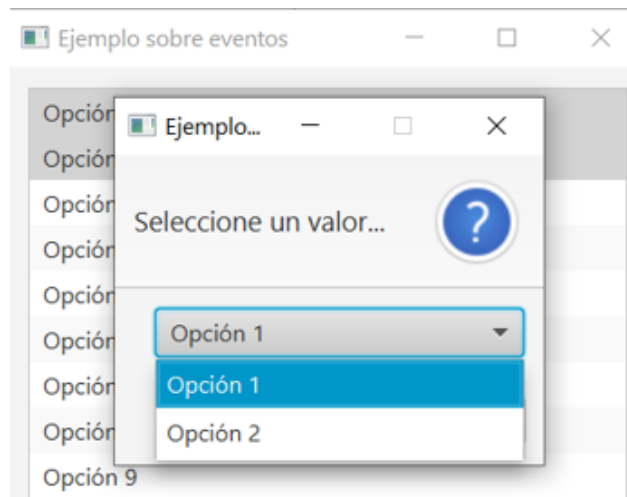
- El listado se crea con los 10 ítems que se pueden observar en la imagen. Se debe configurar para poder seleccionar varios elementos al mismo tiempo.
- El botón está deshabilitado por defecto.

Deberás añadir las funcionalidades que se indican debajo:

- El botón “Mostrar diálogo” se habilita cuando haya algún ítem del listado seleccionado.
- Mientras no haya ningún ítem seleccionado, se mostrará el siguiente **Alert** cuando el usuario deje de posicionar el ratón sobre el listado.



- Cuando haya algún ítem seleccionado, el botón “Mostrar diálogo” tendrá funcionalidad. Al hacer click se muestra el siguiente diálogo de tipo **ChoiceDialog**. El **ComboBox** del diálogo contiene todos los ítems que se han seleccionado en el listado. Por ejemplo, si selecciono “Opción 1” y “Opción 2” serán los ítems que tenga el **ComboBox** para seleccionar.



- Cuando el usuario hace click en “Aceptar” en el diálogo, se volverá a la página principal y el listado tendrá un único ítem seleccionado, que será aquel que se ha elegido en el **ComboBox** de dicho diálogo. Por ejemplo, si selecciono “Opción 1” en el diálogo, si hay otras opciones marcadas en el listado principal se deseleccionarán para que solo aparezca “Opción 1”.

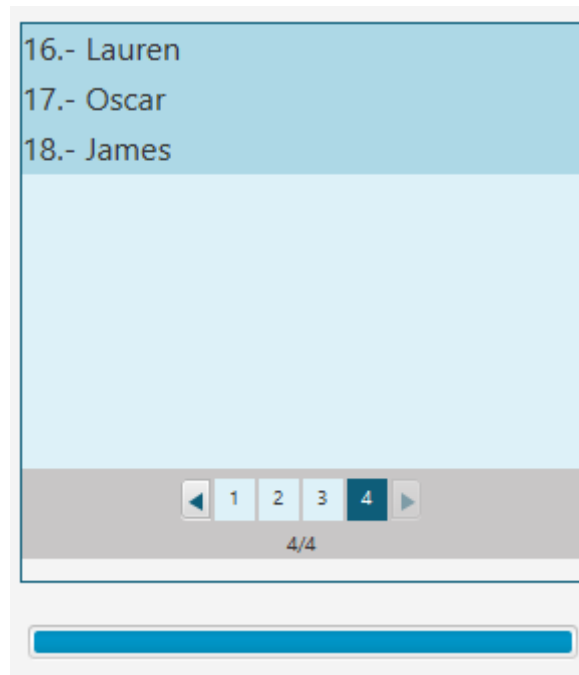
Ejercicio 5

Crea un paquete denominado **pagination** en el cual se incluya un paginador similar al que se indica debajo.



El siguiente enlace te puede resultar de utilidad: <https://www.ic-mouse.net/javafx/control-de-paginacion-pagination-en-tus-apps-fx>

Como puedes observar en la imagen de arriba, la página está asociada a un listado de nombres y se muestran 5 ítems al mismo tiempo (se incluye el orden que ocupan en el listado). Si la última página contiene menos de 5 ítems se mostrarán los restantes.



El número de páginas totales se asigna en la propiedad **setPageCount** del paginador y se calcula en función del número de elementos del listado.

Cada página nueva se crea en una función que recibe como parámetro el método **setPageFactory**. Debes crear un **VBox** con 5 elementos **Label** similar al enlace.

Por último, la barra de progreso se actualiza en el **listener** asociado a la propiedad con el índice de la página seleccionada actualmente. Como la barra de progreso admite valores de 0 a 1, en el ejemplo de la imagen, cuando se selecciona la página 1 tendrá un valor de $1/4 = 0,25$, en la página 2 será de $2/4 = 0,5$ y así sucesivamente.