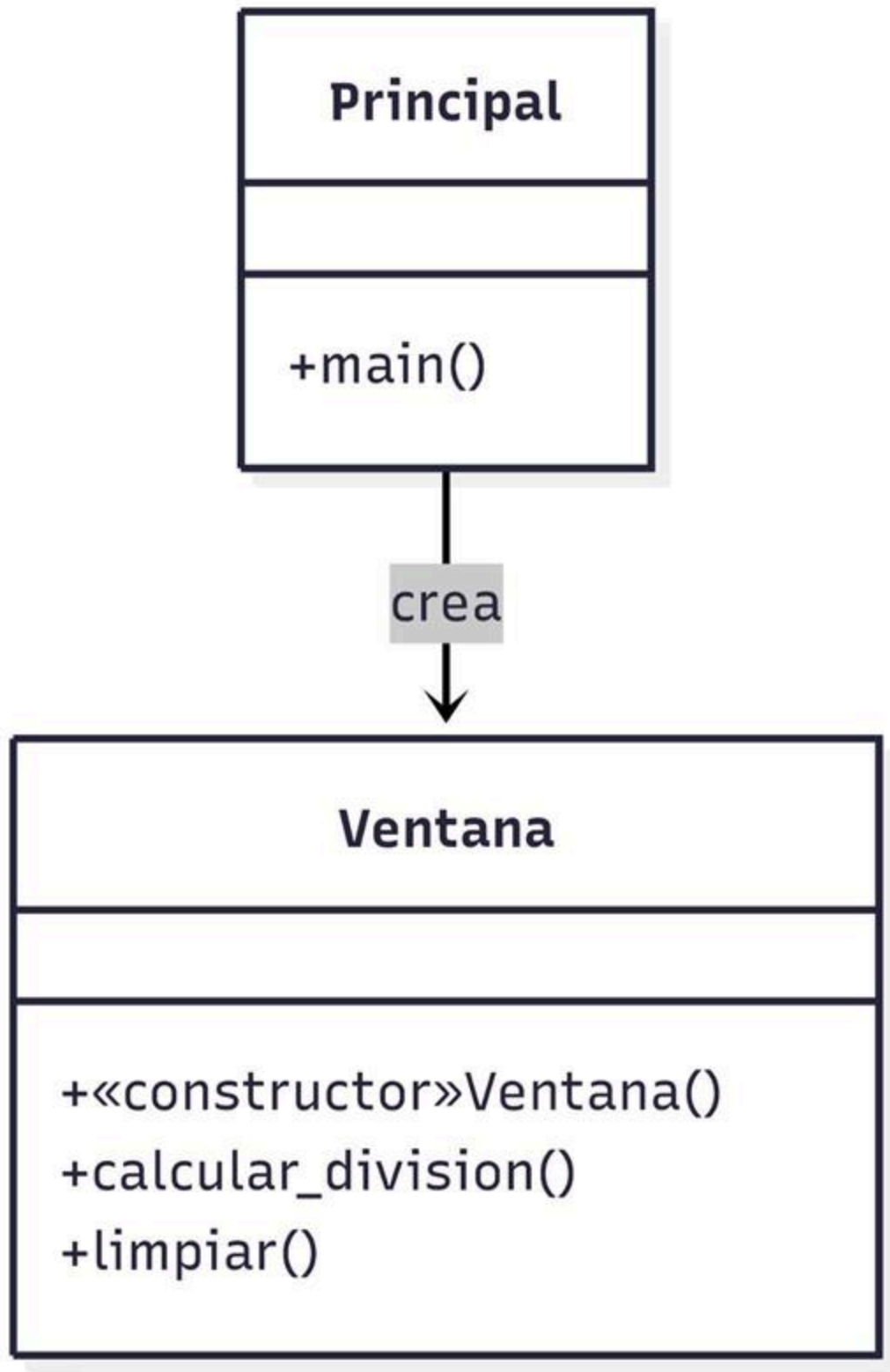


# **Ejercicio 1**

**Enunciado**

**Diagrama de Casos de uso**

**Diagrama de Clases**



## Solución

[Click para ver código fuente](#)

## Ejecución del programa

Sin ingresar un valor

Calculadora


Numerador

Denominador

Resultado

Calcular    Borrar

Error

 Campo nulo o formato de entrada no válido.

Aceptar

**Ingresando solamente un valor no válido**

Calculadora


Numerador

Denominador

Resultado

Calcular

Error

 Campo nulo o formato de entrada no válido.

Aceptar

**Ingresando solamente un valor válido/numérico**

Calculadora


Numerador

Denominador

Resultado

Calcular

Error

 Campo nulo o formato de entrada no válido.

Aceptar

**Ingresando 0 como denominador**

Calculadora


Numerador

Denominador

Resultado

Calcular

Error

 No es posible realizar una división por 0.

### Ingresando valores válidos

Calculadora

Numerador

Denominador

Resultado

Calcular

## **Ejercicio 2**

**Enunciado**

**Diagrama de Casos de uso**

**Diagrama de Clases**

**Principal**

+main()

crea

**Ventana**

+«constructor»Ventana()

+mostrar\_mensaje()



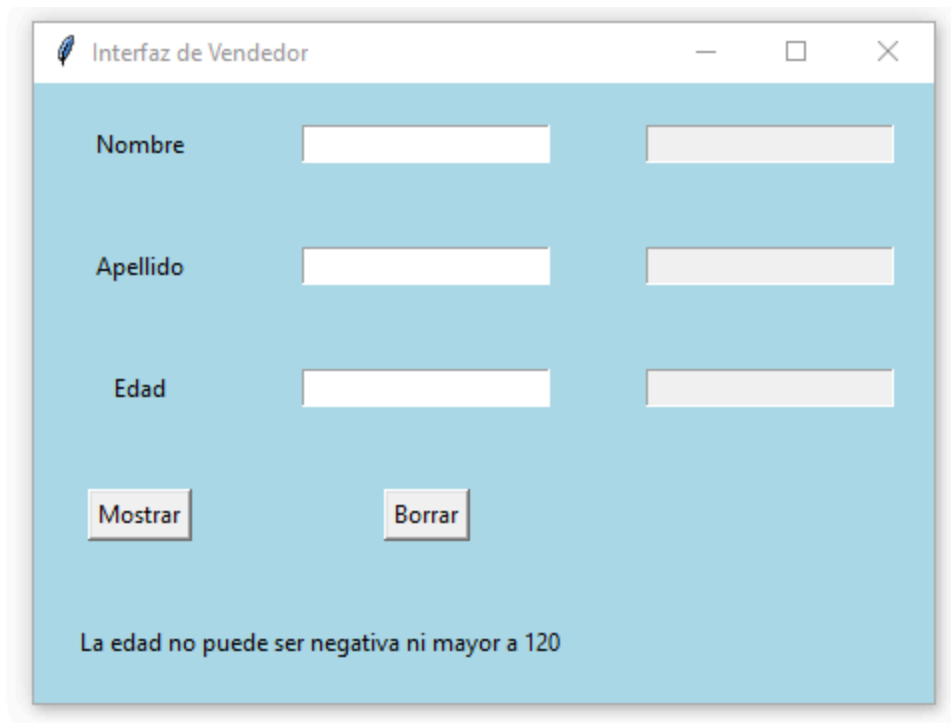
# +limpiar()

## Solución

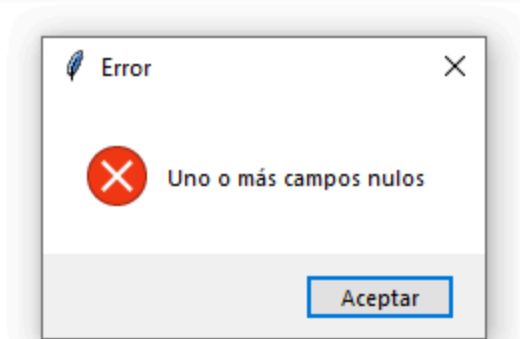
[Click para ver código fuente](#)

## Ejecución del programa

Sin ingresar ningún valor



The screenshot shows a window titled "Interfaz de Vendedor" with a light blue background. It contains three input fields for "Nombre", "Apellido", and "Edad", each with a white text box and a grey placeholder box. Below the fields are two buttons: "Mostrar" and "Borrar". At the bottom, a message reads "La edad no puede ser negativa ni mayor a 120".



Interfaz de Vendedor


Nombre

Apellido


Edad

La edad no puede ser negativa ni mayor a 120

Error

 Campo de edad nulo o formato de edad no válido.

**Ingresando solamente un valor**

 Interfaz de Vendedor

Nombre


Apellido


Edad

Mostrar

Borrar

La edad no puede ser negativa ni mayor a 120

 Error

 Uno o más campos nulos

Aceptar

Interfaz de Vendedor


Nombre

Apellido

Edad

La edad no puede ser negativa ni mayor a 120

Error

 Campo de edad nulo o formato de edad no válido.

**Ingresando un número negativo**

Interfaz de Vendedor


Nombre

Apellido

Edad

La edad no puede ser negativa ni mayor a 120

Error

 La edad no puede ser negativa ni mayor a 120

**Ingresando un número menor a 18 y mayor o igual a 0**

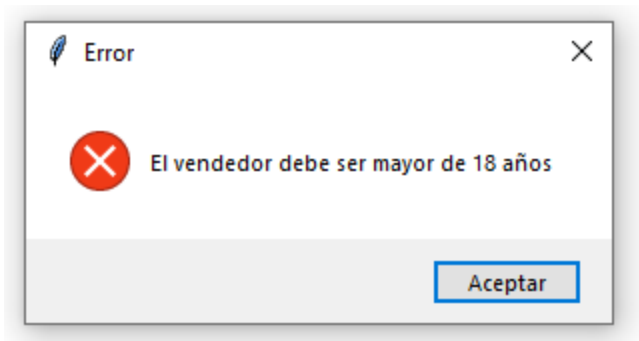
Interfaz de Vendedor

Nombre

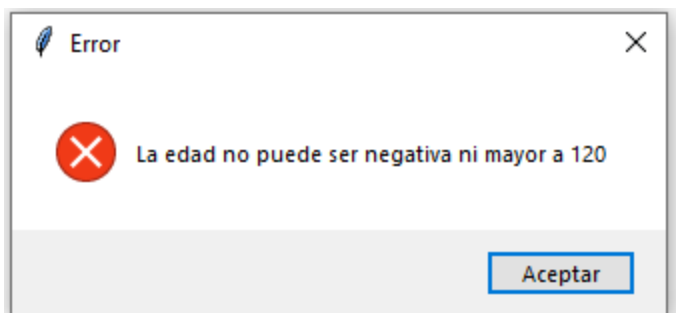
Apellido

Edad

El vendedor debe ser mayor de 18 años



### Ingresando un número mayor a 120

A window titled "Interfaz de Vendedor" with a light blue background. It contains three input fields: "Nombre" with the value "Pedro", "Apellido" with the value "Pérez", and "Edad" with the value "121". To the right of each input field is a corresponding disabled grey field. Below the fields are two buttons: "Mostrar" and "Borrar". At the bottom, there is a text label: "La edad no puede ser negativa ni mayor a 120".


### Ingresando un valor no válido

Interfaz de Vendedor

Nombre	<input type="text" value="Pedro"/>	<input type="text" value="Pedro"/>
Apellido	<input type="text" value="Pérez"/>	<input type="text" value="Pérez"/>
Edad	<input type="text" value="asdsada"/>	<input type="text"/>

La edad no puede ser negativa ni mayor a 120

Error

 Campo de edad nulo o formato de edad no válido.

### Ingresando un valor válido

Interfaz de Vendedor

Nombre	<input type="text" value="Pedro"/>	<input type="text" value="Pedro"/>
Apellido	<input type="text" value="Pérez"/>	<input type="text" value="Pérez"/>
Edad	<input type="text" value="19"/>	<input type="text" value="19"/>

La edad no puede ser negativa ni mayor a 120

# **Ejercicio 3**

**Enunciado**

**Diagrama de Casos de uso**

**Diagrama de Clases**



**Principal**

+main()

crea

**Ventana**

+«constructor»Ventana()

+calcular\_log()

+calcular raiz cuadrada()

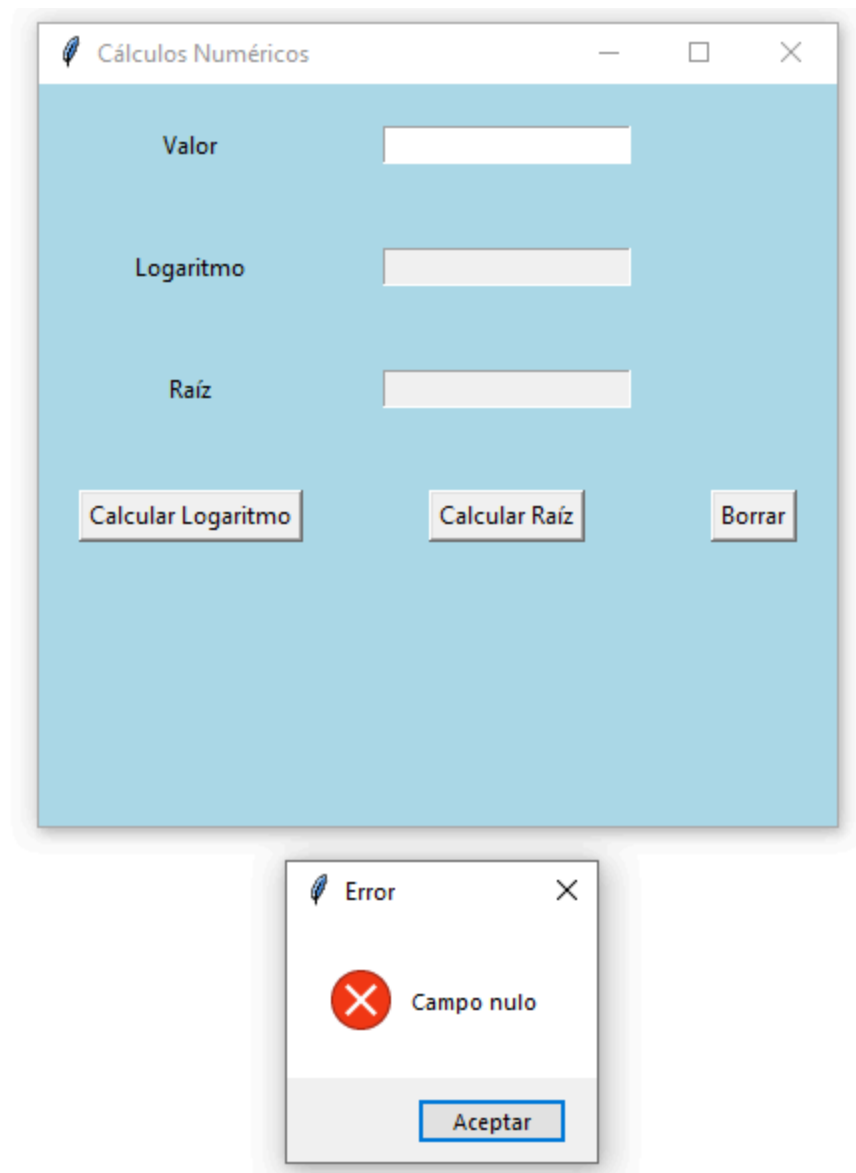
```
calcular_raiz_cuadrada()  
+limpiar()
```

## Solución

[Click para ver código fuente](#)

## Ejecución del programa

Sin ingresar un valor



Ingresando un valor negativo

Cálculos Numéricos

Valor

Logaritmo

Raíz

Error: el valor debe ser un número positivo para calcular el logaritmo

Error: el valor debe ser un número positivo para calcular la raíz cuadrada

**Ingresando 0 como valor**

Cálculos Numéricos


Valor

Logaritmo

Raíz

Error: El valor debe ser distinto de 0 para calcular el logaritmo

**Ingresando un valor no válido**

 Cálculos Numéricos

Valor

sadasd

Logaritmo

Raíz

Calcular Logaritmo


Calcular Raíz

Borrar

Error: el valor debe ser numérico para calcular el logaritmo

Error: el valor debe ser numérico para calcular la raíz cuadrada

### Ingresando un valor válido

 Cálculos Numéricos

Valor

25

Logaritmo

3.22

Raíz

5.0

Calcular Logaritmo

Calcular Raíz

Borrar

## Ejercicio 6.7. Validación de campos (p. 417)

Las excepciones se pueden utilizar para validar el formato de campos de entrada. Por ejemplo:

- Restringir que un campo de entrada de datos sea numérico, alfabético, alfanumérico, etc.
- Restringir la cantidad de caracteres que debe tener un campo de entrada de datos.
- Impedir que se ingresen fechas anteriores a la fecha actual.
- Definir campos obligatorios en un formulario de tal manera que no reciba valores vacíos.
- Definir restricciones especiales de los campos de entrada. Por ejemplo, si es un correo electrónico, que tenga el carácter @.

Para estos casos, si se presentan situaciones que no cumplen dichas condiciones se debe generar la excepción correspondiente.

### Objetivo de aprendizaje

Al finalizar este ejercicio, el lector tendrá la capacidad para definir métodos que realicen validación de campos y generen las excepciones apropiadas en caso de que no se cumplan las condiciones estipuladas en los requisitos de un programa.

### Enunciado: clase EquipoMaratónProgramación

Un equipo de programadores desea participar en una maratón de programación. El equipo tiene los siguientes atributos:

- Nombre del equipo (tipo String).
- Universidad que está representando el equipo (tipo String).
- Lenguaje de programación que va a utilizar el equipo en la competencia (tipo String).
- Tamaño del equipo (tipo int).

Se requiere un constructor que inicialice los atributos del equipo. El equipo está conformado por varios programadores, mínimo dos y máximo tres. Cada programador posee nombre y apellidos (de tipo String). Se requieren además los siguientes métodos:

- Un método para determinar si el equipo está completo.
- Un método para añadir programadores al equipo. Si el equipo está lleno se debe imprimir la excepción correspondiente.
- Un método para validar los atributos nombre y apellidos de un programador para que reciban datos que sean solo texto. Si se reciben datos numéricos se debe generar la excepción

correspondiente. Además, no se permiten que los campos String tengan una longitud igual o superior a 20 caracteres.

- En un método main se debe crear un equipo solicitando sus datos por teclado y se validan los nombres y apellidos de los programadores.

## Solución

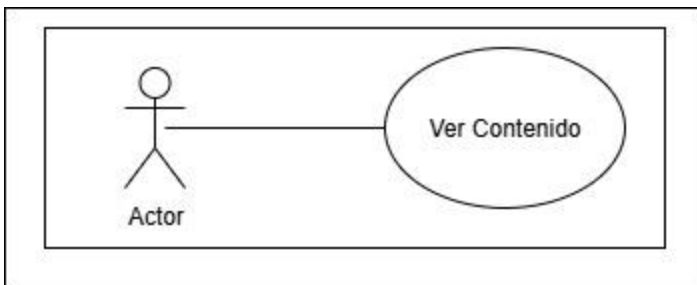
[Click para ver código fuente](#)

### Ejecución del programa

## Ejercicio 3

### Enunciado

### Diagrama de Casos de uso



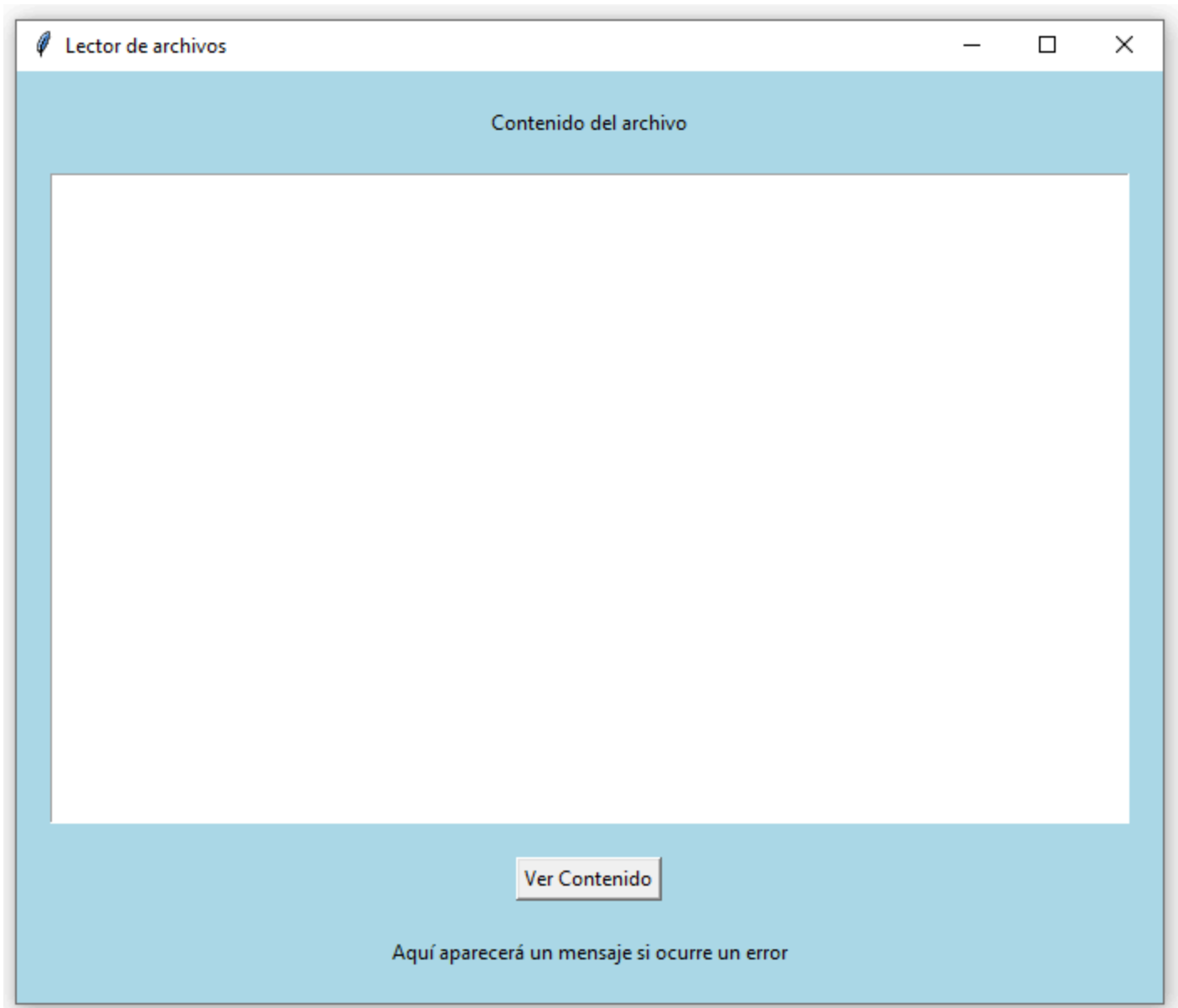
### Diagrama de Clases

## Solución

[Click para ver código fuente](#)

### Ejecución del programa

### Interfaz gráfica



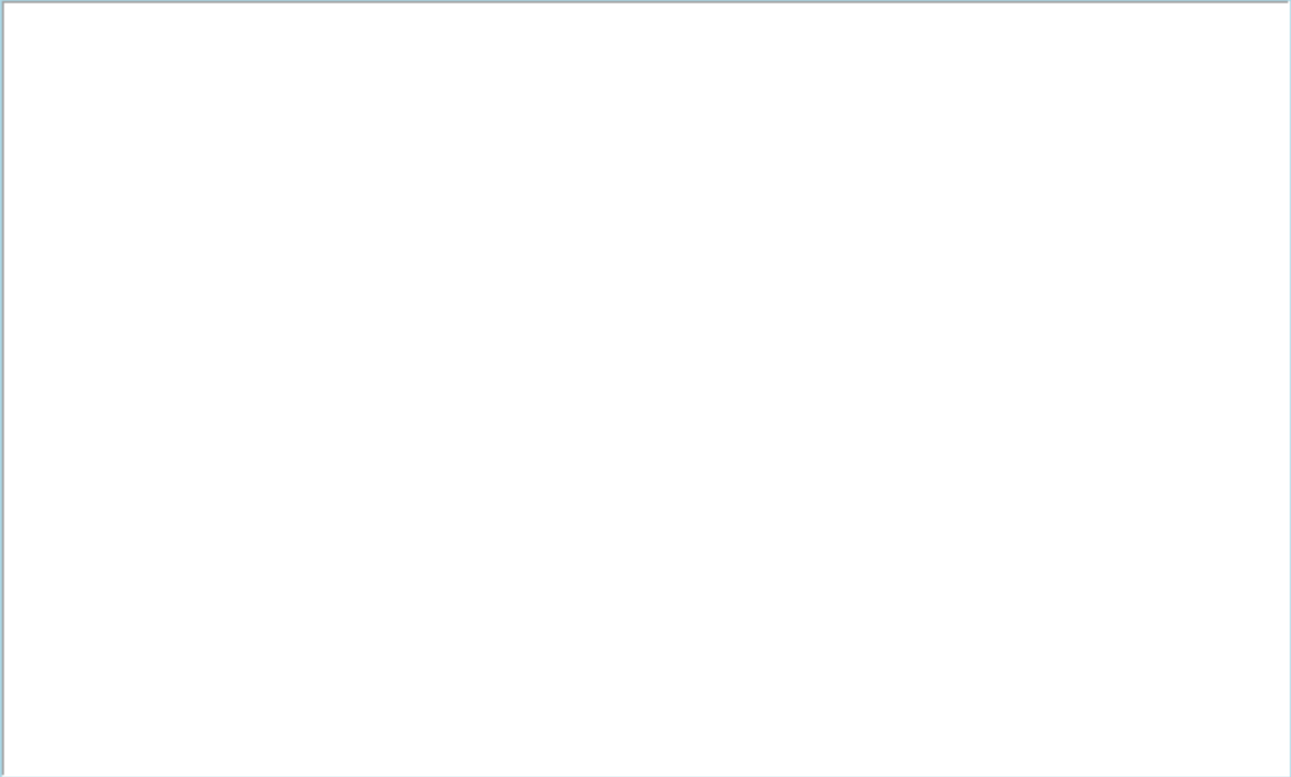
**Cuando el archivo de prueba está en la misma ubicación del trabajo con el código**



**Cuando el archivo de prueba NO está en la misma ubicación del trabajo con el código**



Contenido del archivo



Ver Contenido

No se pudo leer el archivo