



Universidad Técnica Federico Santa María
Elo-329 Diseño y Programación Orientada a Objetos
Proyecto Personal

Profesor: Agustín González.

Documentación

Autor : David Enrique Guzmán Valdivia.

Rol : 9704707-3.

Email : davisgv@elo.utfsm.cl.

Nombre del Proyecto: Cronómetro.

1.- Descripción del problema :

Por lo general la distribución del tiempo entre Expositores de un Seminario o Curso en particular, no es equitativo, y se presta para el abuso o el mal uso de éste por parte de algunos presentadores, en desmedro de los demás. Es por ello que es necesario administrar, controlar y monitorear el tiempo utilizado en presentaciones. El desarrollo de este software busca solucionar el problema implementando un crono metro con alarmas visuales y sonoras.

2.- Análisis del problema.

El escenario donde se presenta el problema de controlar el tiempo es en presentaciones realizadas por alumnos en la universidad, es allí donde se enfoca el desarrollo del problema.

Dado el escenario, existen distintos comportamientos que se desean implementar:

- En primer lugar, el cronómetro debe tener la capacidad de ser configurado, es decir, poder setear el tiempo requerido.
- Existe la posibilidad de pausar una presentación, ello requerirá poder detener el cronómetro para posteriormente continuar.
- El presentador también puede abandonar su presentación debiendo detener el cronómetro ó abortar la cuenta.
- Los expositores necesitan de alarmas visuales para saber el tiempo transcurrido y poder regular la duración de su presentación, además, al finalizar la cuenta se tendrá una alarma sonora.
- Se debe tener la capacidad de poder expandir el tiempo configurado.

3.- **Diseño.**

En la solución del problema se crearon cuatro clases las cuales se describen a continuación:

Class Crono extend JFrame :

Clase principal contiene tres botones, el primero para Correr el Cronómetro “Run Crono”, el segundo para configurarlo “Configuración” y el tercero para abandonar el programa. En esta clase se encuentra el “main”, además contiene un método con el cual una instancia de ConfigCrono puede entregar los parámetros configurados para posteriormente poder ser ingresado a una instancia de RunCrono (tiempo en segundos).

Class ConfigCrono extend JFrame :

Clase que permite la configuración del cronometro, pudiendo ingresar los minutos y segundos deseados, comprueba consistencia en los datos, guarda los datos configurados en un archivo “config.ini” utilizando una Properties. Entrega los datos configurados a una instancia de la clase Crono a través de la función “SetDatos(int tiempo)”. Contiene dos botones, uno para “Guardar” los parámetros ingresados y otro para “Cancelar” y volver a la pantalla principal.

Class RunCrono extends JFrame implements ActionListener:

En esta clase recibe el tiempo configurado a través de la función “IniCrono” y crea una instancia de la clase Tiempo. Con la utilización de un “Timer” y la implementación de ActionListener con su única función **actionPerformed** es posible actualizar cada un segundo la cuenta que lleva la instancia de la clase Tiempo y poder controlar las luces y la barra de progreso implementada.

Se tiene la capacidad de iniciar, detener, continuar, cancelar o expandir la cuenta configurada utilizando botones y campos de texto.

Además existe otro Timer que funciona cuando la cuenta se hace cero y permite dar el efecto de encendido/apagado de la última luz (de izquierda a derecha) la cual es de color rojo.

Clase Tiempo:

Clase utilitaria que recibe de una instancia de RunCrono el tiempo configurado para el cronómetro. Su misión es disminuir el tiempo y entregar un string en formato “minutos:segundos” para poder ser desplegados en el JFrame de RunCrono. Además permite extender el tiempo, entrega el valor actual del cronómetro y el valor inicial configurado.

En la siguiente figura se indica la función que realizan los botones con respecto a los llamados de ventanas (instancias de JFrame) y se presenta el resultado final de la interfaz

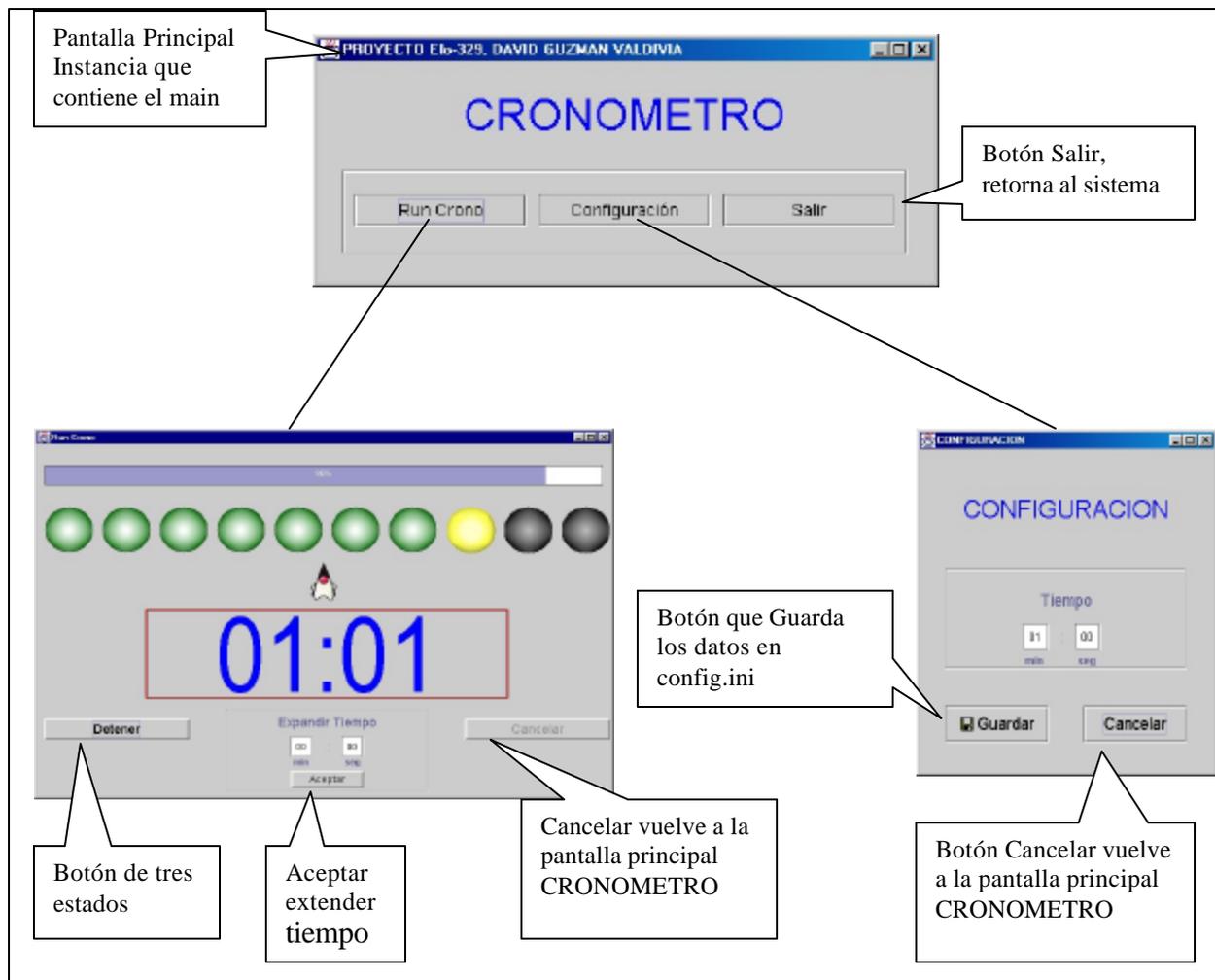


Figura 3.1 Asociación de Botones.

