

**MARIA FERNANDA ACUÑA SALAS**

**uNIVERSIDAD HISPANOAMERICANA**

**Proyecto Programable (Visor de Procesos/Servicios).**

**Bachillerato Ingeniería Informática**

Profesor : Yaser Augusto Castillo Bravo

|  |  |
| --- | --- |
|  |  |

Tabla de contenido

[Informe del proyecto 2](#_Toc486303103)

[Solución Planteada 2](#_Toc486303104)

[Requerimientos Establecidos 2](#_Toc486303105)

[Documentación y estándares de la aplicación 3](#_Toc486303106)

[Interfaz Grafica 3](#_Toc486303107)

[Librerías y Referencias 4](#_Toc486303108)

[Estándares de los Objetos de la Interfaz Gráfica. 5](#_Toc486303109)

[Variables Destacadas 5](#_Toc486303110)

[Clases o Métodos 6](#_Toc486303111)

[Funciones 9](#_Toc486303112)

[Resultados Obtenidos 14](#_Toc486303113)

[Requerimientos No Cumplidos 16](#_Toc486303114)

Informe del proyecto

Solución Planteada

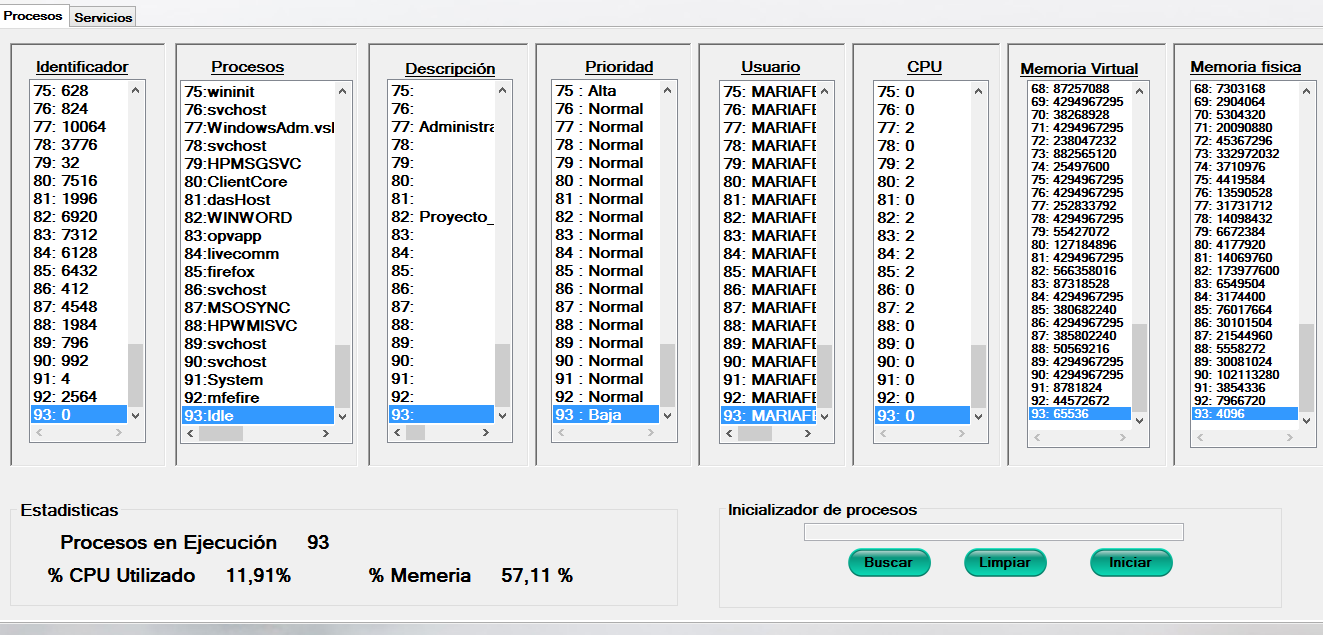
Para la realización de este proyecto se planteó desarrollar una aplicación en el lenguaje de programación C# utilizando Windows Form. En la cual se pobra consultar los procesos y servicios locales que se estén ejecutando en el momento, además abra una opción de iniciar un proceso nuevo o detener un proceso en ejecución. También abra una opción para iniciar un servicio local o suspender el servicio.

Requerimientos Establecidos

* Se puede consultar los procesos/Servicio (Identificador, Nombre, Descripción).
* En el caso de los procesos se pobra consultar los recursos utilizados.
* Se debe brindar la posibilidad de iniciar, terminar procesos.
* En el caso de los servicios se brindará la posibilidad de pausarlos.
* Se debe brindar la opción de cambiar la prioridad de un proceso.
* Establecer la afinidad de un proceso.
* Se podrá consultar un resumen con los porcentajes de utilización de los principales recursos (Cantidad de procesos en ejecución, Porcentaje de CPU utilizado, Porcentaje de memoria utilizada).

Documentación y estándares de la aplicación

Interfaz Grafica



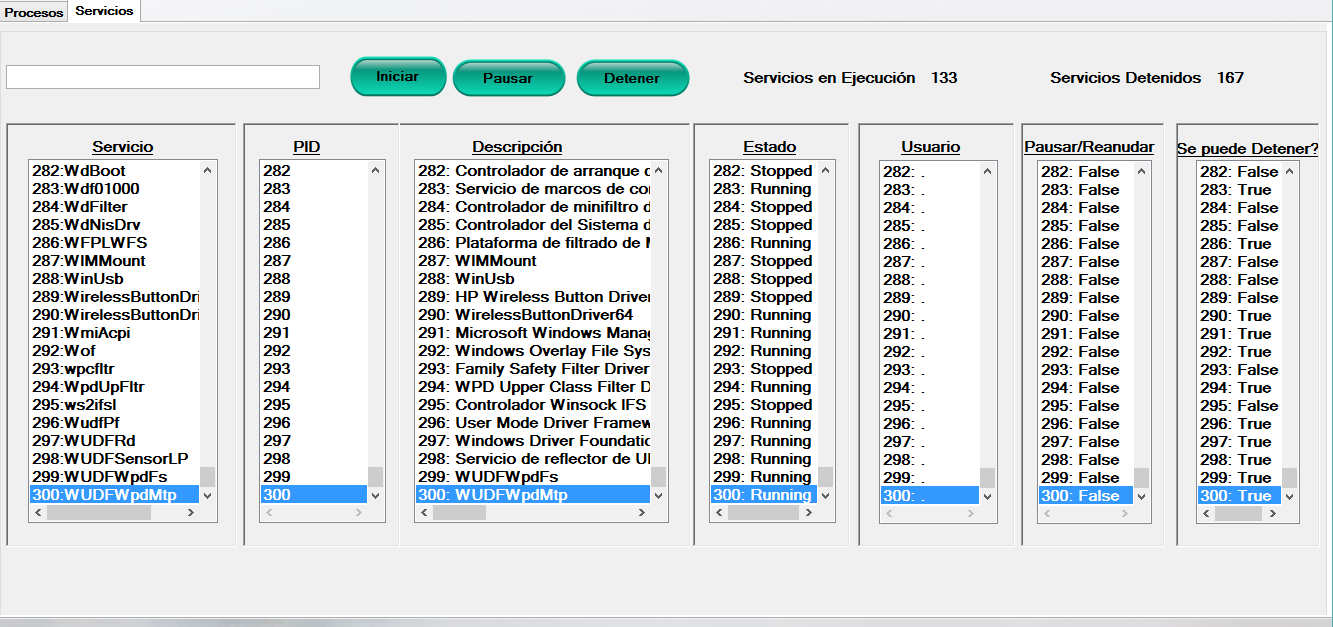
En esta interfaz se puede ver varios listBox los cuales contiene los datos de los procesos tales como prioridad, nombre, ID, memoria que utiliza, etc.…

Además de los datos estadísticos como el número de procesos en ejecución, el porcentaje de memoria utilizada y el porcentaje de CPU utilizado.

También hay tres botones y dos menús contextuales los cuales son:

* Buscar: busca el proceso que quiere iniciar.
* Limpiar: limpia el Textbox de búsqueda.
* Iniciar: Inicia el proceso.
* ContxtMenu: Actualiza y sale de la aplicación.
* ContxtProceso: Simula un cambio de prioridad.

Nota: Debe posicionarse dentro de la lista que contenga los nombres de los procesos para acceder a este sub menú, debe de seleccionar algún proceso de la lista para realizar cualquier acción del submenú.



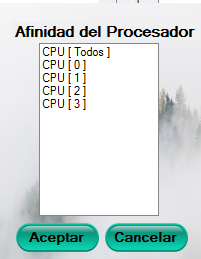
Esta segunda parte de la interfaz consta igual que la anterior con múltiples Listbox los cuales contiene los datos de los servicios.

También se puede ver los datos de cuantos servicios están corriendo y cuales estos detenidos.

Consta con tres botones y un contextMenu(el cual tiene las mismas acciones que los botones):

* Reanudar: el cual inicia el servicio. (Iniciar)
* Pausar: que suspende el servicio.
* Detener: Finaliza el servicio que se esté corriendo.

Nota: Debe seleccionar algún elemento de la lista Servicios para pausar el servicio.



En esta interfaz muestra los Kernel que contiene su computador y da la opción de estable ser el proceso en uno en específico o en todos.

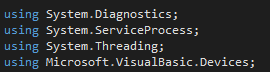
Está compuesto por un listBox, dos botones y un label.

Aceptar: establece la nueva afinidad del proceso.

Cancelar: cierra la ventana.

Librerías y Referencias

Se utilizaron múltiples librerías para realizar esta aplicación, las cuales son:



Además de la referencia:





Estándares de los Objetos de la Interfaz Gráfica.

Cada objeto que compone la interfaz gráfica tendrá un identificador antes del nombre indicando de que tipo es.

**Estándares**

* **Label-> lb**
* **Panel->pl**
* **Textbox-> txt**
* **ListBox-> list**
* **Button-> btn**
* **TabControl->tab**
* **GroupBox-> gp**
* **TabPag-> tabPag**
* **Timer->trm**
* **ContextMenuStrip-> Contxt**
* **ToolStripMenuItem-> Tool**

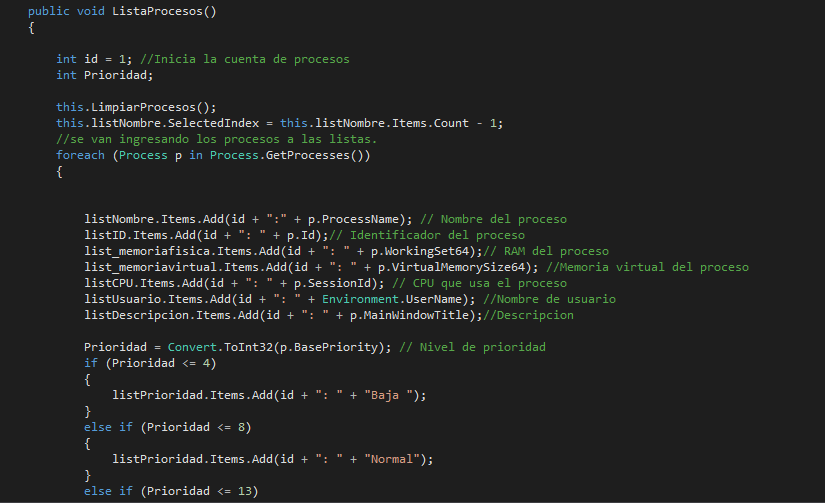
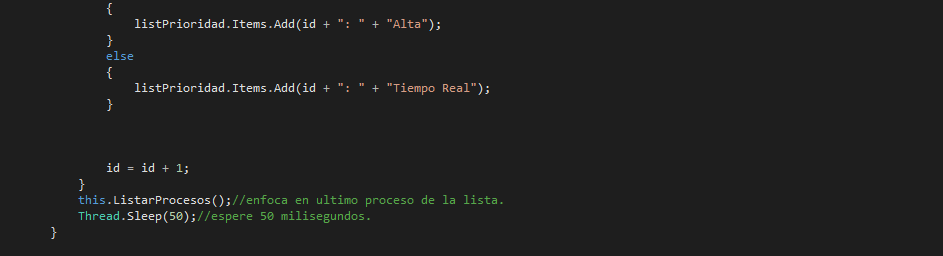
Variables Destacadas

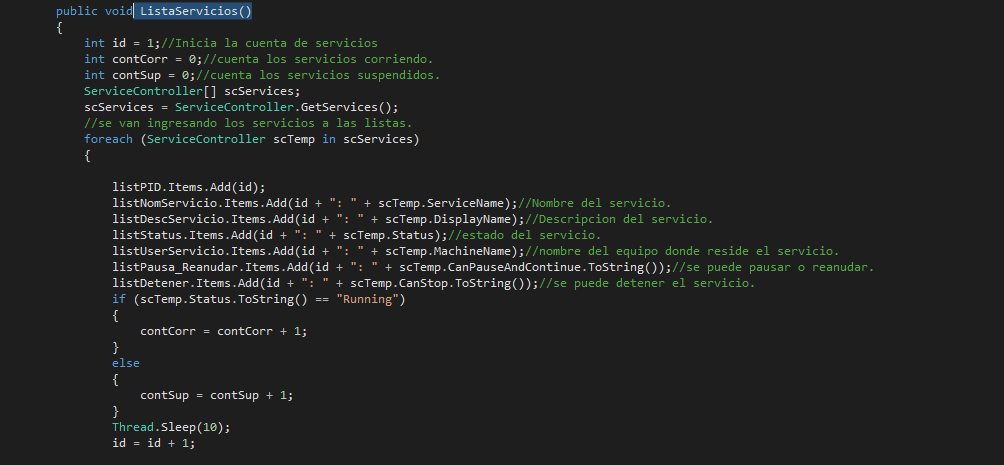




Clases o Métodos

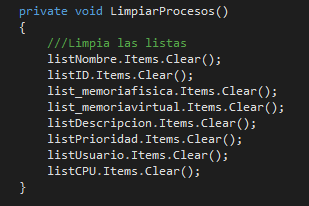
ListaProcesos() o ListaServicios(): estos métodos ingresan los datos de los procesos y servicios a los listbox.

**ListaProcesos()**

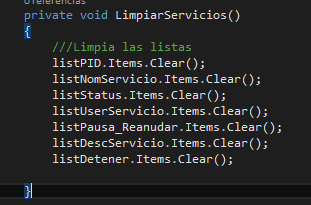
**ListaServicios():**

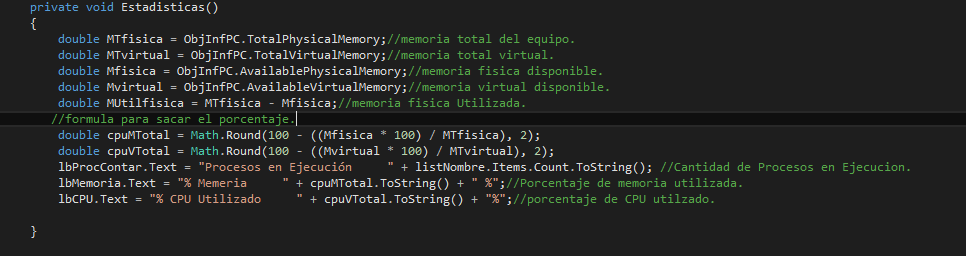
**LimpiarProcesos() y LimpiarServicios():** Elimina los datos de las listas.

**LimpiarProcesos()**

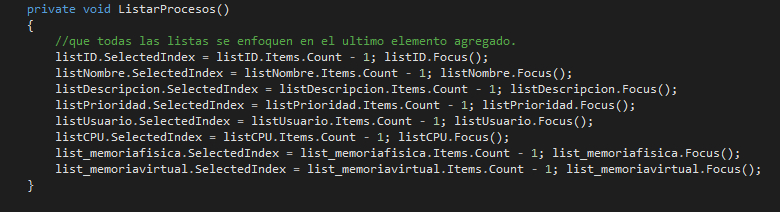


**LimpiarServicios()**

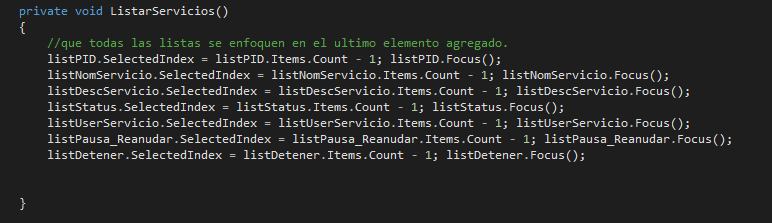


**Estadísticas():**Muestra los datos estadísticos, como lo son el número de procesos, memoria utilizada…

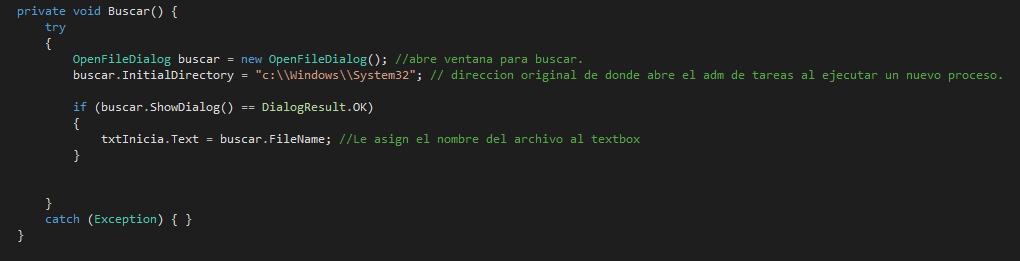
**ListarProcesos() y ListarServicios():**que todas las listas se enfoquen en el último elemento agregado.

**ListarProcesos()** 

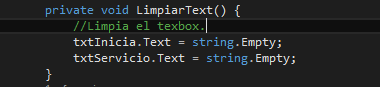
**ListarServicios():**



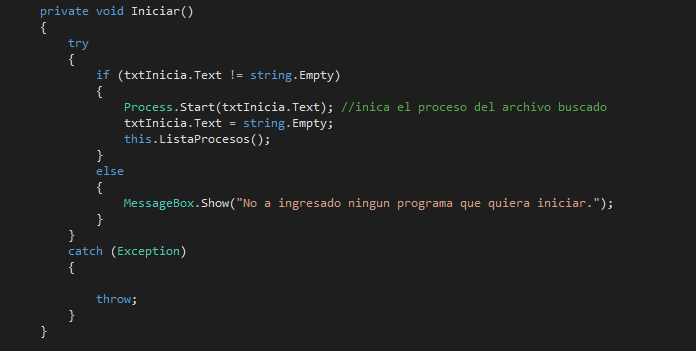
Funciones

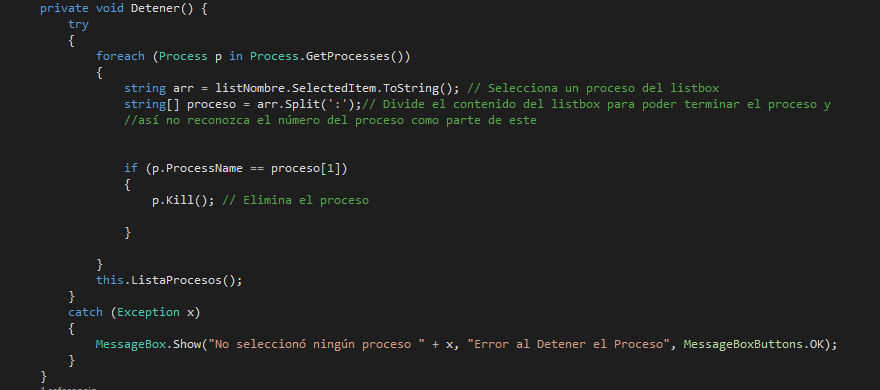
**Buscar():** se encuentra dentro del botón (btnbusc) . Abre una ventana donde se debe de seleccionar el programa que quiere abrir, al seleccionarlo y aceptar ingresa la dirección a textbox(txtInicia).

**LimpiarText():** limpia los textbox de la aplicación.

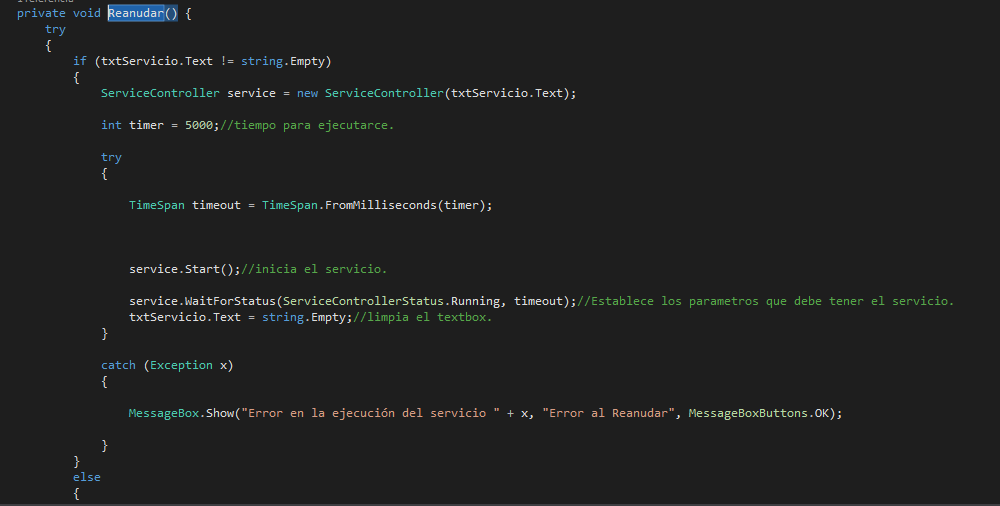
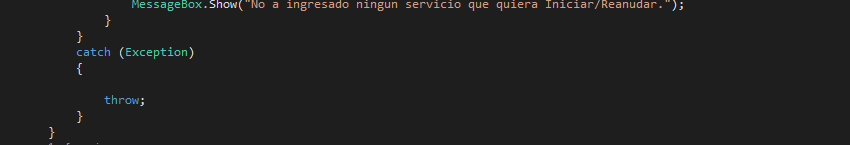


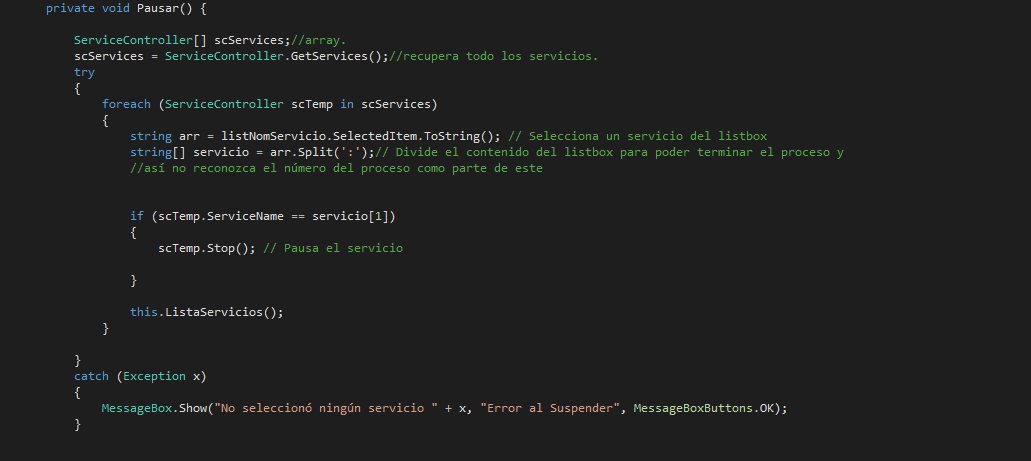
**Iniciar():** verifica que el txtInicia no este vacío luego inicia el proceso que tenga la dirección que contenía el textbox, limpia el texbox y actualiza. se encuentra en el botón(btnInicio).



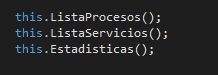
**Detener():** para acceder a esta funcion se debe de seleccionar algun elemento de la lista (listNombre), a este dato se le quita el id del numero de elemento que representa en la lista. Luego se evalua que el dato extraido exista como un proceso y se finaliza el proceso.

**Reanudar():** es una función muy parecida a la de iniciar. En la cual verifica que txtServicio no esté vacío, ingresa el dato que contenía txtServicio en una variable de tipo ServiceController, también se estable una variable de tipo int que contendrá el tiempo en el que se realizará el servicio. Se inicia el servicio y se establece que está en ejecución (Running) y el tiempo.

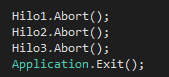


**Pausar():** para acceder a esta funcion se debe de seleccionar algun elemento de la lista (listNomServicio), a este dato se le quita el id del numero de elemento que representa en la lista. Luego se recorre los datos de los servicios por medio de un foreach, evalua que el dato extraido exista como un servicio y si se encuentra se suspende el servicio.

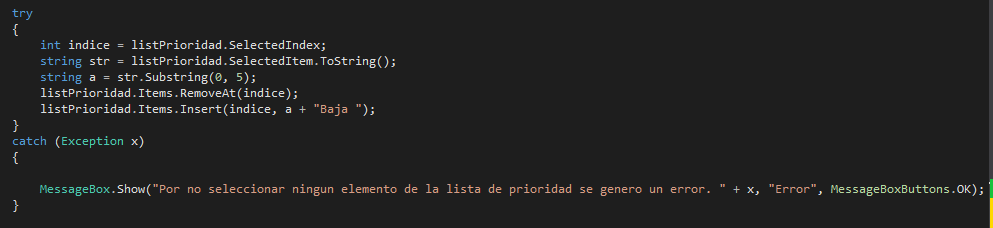
**ActualizarTool:** Actualiza los datos.



**SalirTool:** Sale de la aplicación.

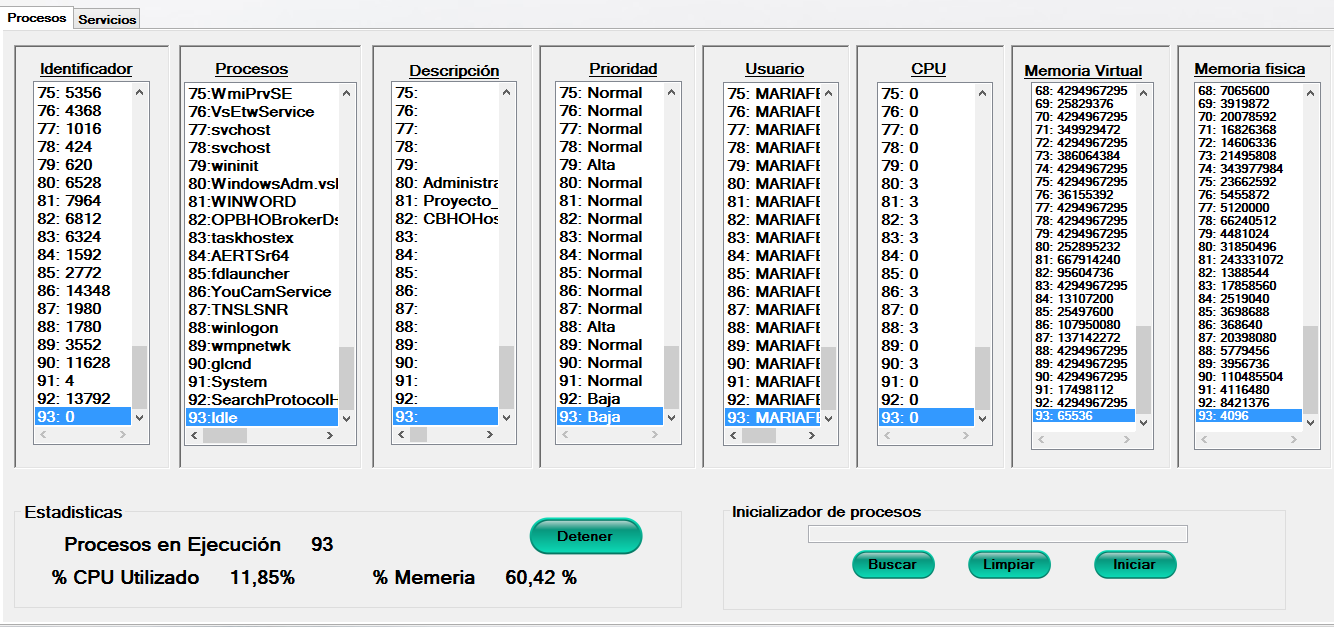


**tiempoRealTool, altaTool, normalTool, bajaTool:** Simulan un cambio de prioridad.



Resultados Obtenidos

Como resulta del desarrollo de este proyecto se obtuvo una aplicación la cual se puede consultar los datos de los procesos y servicios que estén ejecutándose en el equipo. Además de cuantos procesos hay, cuantos servicios estén corriendo y cuantos están suspendidos, cuanto porcentaje de memoria física y virtual se está utilizando.

También se puede iniciar un nuevo proceso o finalizar un proceso que este en ejecución, y en el caso de los servicios se puede suspender(pausar) o reanudar.

