



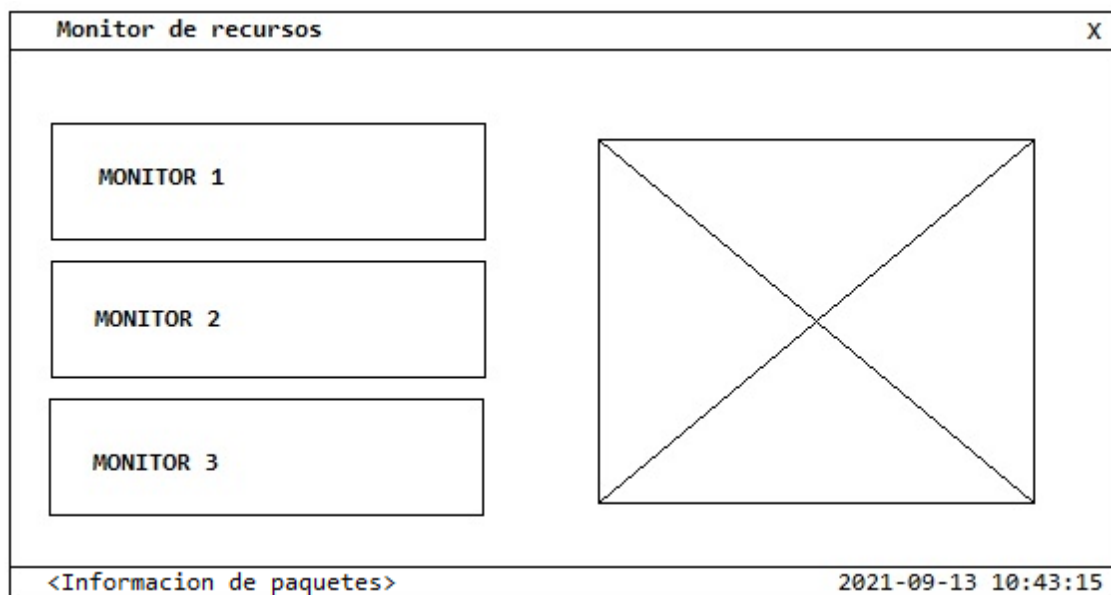
# Proyecto 1

## Monitor de recursos de una PC en tiempo real

Debe de diseñar y codificar una aplicación en Python, utilizando la librería *tkinter*, *matplotlib*, *numpy*, *psutil*, *datetime* y las que considere pertinentes, que registre los eventos en tiempo real de una PC y mostrarlos en un formato gráfico atractivo.

Su aplicación debe de monitorear los eventos de forma gráfica (uso del CPU, uso de la memoria virtual, uso del disco duro), mostrar un gráfico en línea de uno de los recursos (por ejemplo, uso del CPU) y los contadores de paquetes enviados y recibidos por la interfaz de red. Así también, debe de mostrar la hora y fecha del sistema de forma continua.

### Wireframe



### Detalles del wireframe

#### MONITOR 1 -3

Monitor 1: Información sobre el CPU (número de núcleos de procesador, así como el porcentaje de uso en tiempo real).

Monitor 2: Información sobre el uso de la memoria RAM del equipo (capacidad instalada, uso).

Monitor 3: Información sobre el uso de disco duro (capacidad, disponibilidad).

Todo esto se muestra utilizando unos widgets tipo Label, así como widgets Progressbar que muestran de forma gráfica la información.

### GRÁFICA

Se debe de mostrar una gráfica en tiempo real de los cambios en el uso del CPU en el tiempo. En el eje x no se debe de mostrar ninguna indicación, mientras que en el eje y se mostrará la

información de uso de manera porcentual. La gráfica debe de trasladarse de derecha a izquierda, donde los datos más recientes se van mostrando por el lado derecho.

### STATUS BAR

El Status Bar debe mostrar información sobre los paquetes entrantes y salientes en la interface de red en tiempo real. Así también, en el borde derecho, debe mostrarse la fecha y hora del sistema, y esta debe ir actualizándose a nivel de segundos.

### LIBRERÍA psutil

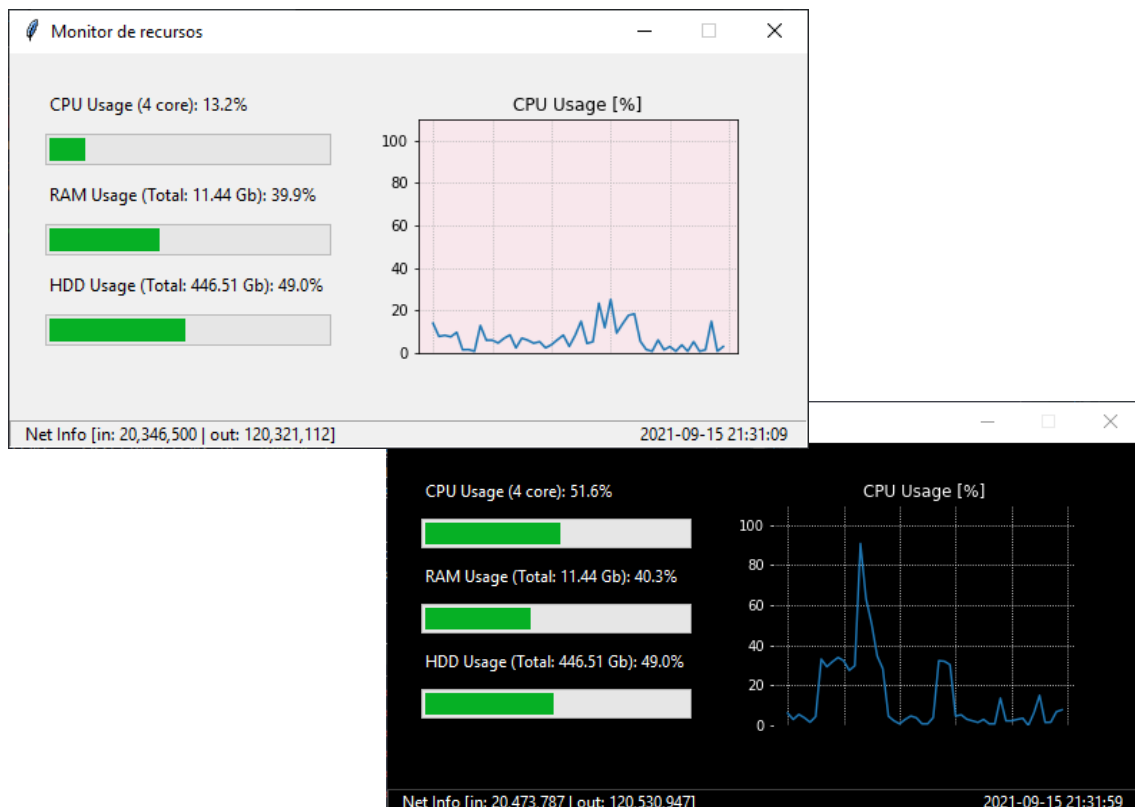
La librería *psutil* permite obtener la información necesaria para la construcción del monitor de recursos. Se adjunta un código (*system\_info.py*) que muestra información en la consola (presione Ctrl+Enter para cancelar la ejecución) que podrá trasladar a la aplicación. Se muestra la captura de la aplicación:

-----time-----	---cpu---	---memory---	---disk---	-----net-----
2021-09-15 20:08:48	4core 41.7%%	11.44G 37.2%%	446.51 49.0%%	in:11776626 out:7656435

Figura 1 – Consola con los resultados del script *system\_info.py*

### Ejemplos de diseño

Se muestran algunos ejemplos de diseño para que los pueda usar como referencia:



### Entregables

Debe de entregar los siguientes archivos al Aula Virtual en los plazos especificados:

- Script con la solución del monitor (*cpu\_monitor.py*). Debe ser un script que siga la metodología de la programación orientada a objetos. La aplicación será una clase con diferentes métodos para el control de las funcionalidades. (La ventana de la aplicación debe de ser pequeña, no debe de ocupar más que la tercera parte del monitor. Observe las proporciones en los ejemplos de diseño). Debe ser un archivo PY.
- Archivo JPG con la captura de la ejecución del monitor en su equipo (*monitor.jpg*), donde se pueda mostrar el alcance de su solución.

### Rúbrica de evaluación

La rúbrica de evaluación se encuentra en el AV.