

## Turno 3

### Consignas Generales.

Se pide desarrollar un programa en Python para el enunciado que sigue. El programa obligatoriamente deberá plantearse como un **proyecto** que contenga al menos **dos módulos** (uno para la definición del tipo de registro y las funciones para gestionarlo, a criterio del estudiante) y otro módulo deberá contener el programa principal que obligatoriamente debe ser planteado en base a un menú de opciones y con funciones para toda situación posible. También debe incluir el control de ejecución del módulo principal con la variable `__name__`. Al finalizar su desarrollo, cada estudiante debe comprimir la carpeta del proyecto y subir ese archivo comprimido a su casillero de entrega para el Parcial 3 a través del Aula Virtual. El proyecto **debe** ser desarrollado exclusivamente usando el IDE PyCharm.

### Enunciado:

Una empresa de tecnología desea desarrollar un software de monitoreo de los sistemas a los que da soporte. El programa debe permitir registrar los errores que los distintos sistemas arrojan para permitir un análisis posterior. Por cada **Error** se tiene un código de error (un número entero), un número de sistema (un número entero que indica el sistema que ha fallado), un mensaje de error (un string), un valor entre 0 y 23 que indica en qué rango de hora se produjo la falla, y la cantidad de segundos que el sistema ha quedado fuera de línea a causa del error (un número entero). Se desea almacenar la información de  $n$  errores en un arreglo de registros (la cantidad  $n$  se debe cargar por teclado). Se pide desarrollar un programa en Python controlado por un menú de opciones y que posea como mínimo dos módulos, que permita gestionar las siguientes tareas:

- Cargar el arreglo pedido con los datos de los  $n$  errores. Debe validar o asegurar que el código de error esté entre 1000 y 5000, y que el resto de los datos sean siempre correctos. Puede hacer la carga en forma manual, o puede generar los datos en forma automática (con valores aleatorios). Pero al menos una debe programar. Pero si hace carga automática, todos los campos deben ser cargados así (no combine ambas técnicas en la carga de un registro). No se requiere validar que los códigos de error sean diferentes.
- Mostrar los datos de todos los errores cuya cantidad de segundos fuera de línea esté entre los valores  $s1$  y  $s2$  (ambos incluidos) que se cargan por teclado, y ordenados de menor a mayor por código de error. Indique al final la cantidad de errores mostrados en este listado.
- Determinar cuántos errores se produjeron en el rango de cada una de las horas posibles (24 contadores). Mostrar todos los conteos que sean diferentes de cero. Al final de este listado, mostrar cual fue la hora en la que se produjo la mayor cantidad de errores.
- Determinar si existe un error cuyo número de sistema y su descripción sean iguales a valores  $num$  y  $a$   $des$  que son ingresados por teclado. Muestre sus datos en caso de encontrarlo, o informe con un mensaje si no existe. Debe mostrar los datos del primero que encuentre, y detener la búsqueda (sin importar si hay más de un registro que cumpla el criterio pedido).

### Criterios generales de evaluación.

- Desarrollo del programa completo, *incluyendo (entre otros) una función principal, menú de opciones, control de ejecución del script principal y dos o más módulos*: **[máximo: 8% del puntaje]**
- Desarrollo correcto del ítem 1: **[máximo: 20% del puntaje]**
- Desarrollo correcto del ítem 2: **[máximo: 22% del puntaje]**
- Desarrollo correcto del ítem 3: **[máximo: 24% del puntaje]**
- Desarrollo correcto del ítem 4: **[máximo: 26% del puntaje]**
- Para aprobar el parcial, el alumno debe llegar a un *porcentaje de al menos 55% del puntaje máximo*, pero **obligatoriamente** debe estar desarrollado el programa, funcionando, operativo e incluyendo dos o más módulos.