
Primer Parcial de Programación 1

06/10/2021

Tener en cuenta que:

- La duración del parcial es de 3 horas.
- Se debe realizar en computadora, para este fin se deberá seguir los siguientes pasos:
 1. Crear una carpeta dentro de su computadora de nombre `primerparcial2021`.
 - En caso de realizar el parcial en una computadora del laboratorio usar la carpeta Z.
 2. Descargar del Moodle el archivo **`primer_parcial_2021.py`** que se encuentra dentro de la carpeta "Parciales Anteriores" dentro de "Primer Parcial"
 3. Mover archivo a carpeta "`primerparcial2021`" creado en paso 1.
 4. Abrir el Visual Studio Code y colocar en el menú la opción File->Open ir a la carpeta creada `primerparcial2021` y poner Open. Esto debería abrir la carpeta y mostrarle el modulo `primer_parcial_2021.py` a la izquierda para seleccionar.
 5. Dentro de Visual Studio Code seleccionar en el menú la opción Terminal->New terminal
 6. En la terminal abierta (abajo) colocar el siguiente comando para verificar que todo esta correcto: `python3 primer_parcial_2021.py`. El resultado esperado es:
 - Error Prueba 1 crear actividades
 - Error Prueba 2 buscar actividades
 - Ok Prueba 3 buscar actividades
 - Ok Prueba 4 consultar aforo sin actividad existente
 - Error Prueba 4.5 consultar aforo actividad existente
 - Ok Prueba 5 consultar reserva sin actividad existente
 - Error Prueba 6 consultar reserva con actividad existente
 - Error Prueba 7 aforo disponible post reserva
- Culminado las horas del parcial se deberá entregar a través del Moodle en la tarea "Entrega Primer Parcial". Se espera que se suba el archivo "**`primer_parcial_2021.py`**" donde coloco el código fuente de la solución.
- Tenga en cuenta que no se corrige código con errores del interprete de Python.
- El parcial es individual, se puede usar material.
- Se contestan dudas solo en la primer hora y media del parcial.

Ejercicio 1 - Practico (Tiempo recomendado 3:00h)

Una empresa de software que realiza soluciones para operadores turísticos, desea realizar una solución que permita controlar el aforo en actividades turísticas. Para ello se desea realizar una primer prueba de concepto que permita controlar aforo por hora de actividades que ofrecen los operadores. Por ejemplo, se tiene la actividad turística "Jardín Japonés" y el mismo dispone un aforo máximo por hora de 20 personas. El objetivo es que los turistas antes de visitar la actividad reserven para cierto día y hora y verifiquen si efectivamente tienen cupo disponible, en caso de no tener cupo deberán elegir otra hora u otro día.

Teniendo esto en mente, se desea realizar un prototipo que valide los algoritmos que van a ser necesarios para llevar este proyecto en marcha. Para acotar la realidad de esta primer versión, el sistema va a manejar reservas de un mes (no importa cual) y las reservas van a ser anónimas.

Para la implementación se plantean las siguientes estructuras que van a ser usadas por los distintos algoritmos:

- **`mat_actividades`**, las actividades ofrecidas por operadores turisticos se van a representar con una matriz (arreglo de dos dimensiones), donde cada fila representa una actividad registrada para control de aforo. Las columnas que debería tener esta matriz son:
 - **`nombre`**, representa el nombre de la actividad que lo identifica
 - **`hora_apertura`**, es la hora en formato 0 a 23 que abre la actividad
 - **`cantidad_horas`**, es la cantidad de horas que la actividad esta disponible
 - **`personas_por_hora`**, es el aforo máximo por hora

Un ejemplo de matriz con dos actividades sería `[["Parque Rodó", 14, 8, 50], ["Jardín Japonés", 12, 5, 20]]`. Esto indica que la actividad Parque Rodó, abre a las 14 horas, esta abierto por 8 horas y tiene

un aforo de 50 personas por otro lado el Jardín Japonés, abre al público a las 12, esta abierto por 5 horas y tiene un aforo de 20 personas por hora.

- **mat_reservas**, las reservas que se realizan de una actividad se van a representar con una matriz, donde cada fila representa una reserva realizada de una persona a una actividad. Las columnas que debería tener esta matriz son:
 - **nombre_actividad**, contiene el nombre de la actividad
 - **día**, contiene el día para la cual la reserva aplica, valores validos son de 1 a 30. (considerar que todos los meses tienen 30 días)
 - **hora**, contiene la hora de la reserva, valores validos de 0 a 23.
 - **cant_personas**, cantidad de personas de la reserva

Considerado estas estructuras se debe implementar las siguientes operaciones, dentro del módulo `primer_parcial_2021.py`. El mismo dispone de los cabezales de las operaciones para que sean completados y un conjunto de pruebas en el método principal. Las operaciones a implementar son:

- **buscar_actividad_por_nombre(mat_actividades, nombre)**
 - Esta operación deberá buscar la actividad por nombre en la matriz `mat_actividades` y retorna la fila que representa la misma. En caso de no encontrar la actividad con el nombre pasado devuelve el vector vacío `[]`.
- **crear_actividad(mat_actividades, nombre, hora_apertura, cantidad_horas, cant_personas_por_hora)**
 - Esta operación deberá crear una actividad para la cual se desea controlar aforo. La `hora_de_comienzo` es un numero que va de 0 a 23, `cantidad_horas` es la cantidad de horas que esta disponible la actividad y tiene que ser mayor a 1 y `cant_personas_por_hora` tiene que ser mayor que cero e indica el aforo máximo de la actividad por hora.
 - Por ejemplo si quisiéramos crear la actividad "Parque Rodo" que abre a las 14 y esta abierto por 8 horas con un aforo de 50 personas x hora, realizaríamos la siguiente llamada: `crear_actividad("Parque Rodo", 14, 8, 50)`
 - En caso de que la actividad ya exista, se debe actualizar los datos `hora_apertura`, `cantidad_horas` y `cant_personas_por_hora`.
 - Retorna `True` si se logra crear la actividad `False` en caso contrario. La actividad no podría crearse si no se cumple alguna precondición, como por ejemplo que la `hora_apertura` no este comprendida entre 1 y 23.
- **consultar_aforo_disponible(mat_actividades, mat_reservas, nombre_actividad, dia, hora)**
 - Esta operación retorna el aforo disponible para la actividad, día y hora indicados. Es importante que no solo considere el aforo máximo para la actividad por hora si no también las posibles reservas existentes que disminuye el aforo disponible.
 - En caso que la actividad no este registrada se debe retornar el valor -1.
- **reservar_actividad(mat_actividades, mat_reservas, nombre_actividad, dia, hora, cant_personas)**
 - Brinda la posibilidad de realizar una reserva para una actividad en un día, mes y hora.
 - En caso de no haber cupos disponibles, no debe realizarse la reserva.
 - Se debe retornar `True` si la reserva fue realizada con éxito, `False` en caso contrario.