



Primera Actividad de Evaluación

GEI

Grau en Enginyeria Informàtica

**Aitor Corchero
Raul Ariño
Ian Palacin**

Dept. Informàtica i Enginyeria Industrial



ASIGNADOR DE MESAS DE UN RESTAURANTE

1 Enunciado

Un restaurante, ante la gran demanda de comensales, necesita de un algoritmo de gestión de las mesas dentro de su local. Los comensales, que hacen reserva en dicho local, tienen la preferencia de reservar lo más cercanamente posible de la cocina con el fin de garantizar que los platos les lleguen en las condiciones más óptimas en el menor tiempo posible.

Es por ello que, si el restaurante tiene “N” mesas siendo la “mesa 1” la más cercana y la “mesa N” la más lejana, los comensales a la hora de realizar la reserva quieren una mesa lo más cercana a la “mesa 1”. Además, hay que tener en cuenta que no tendremos en consideración el número de comensales a la hora de realizar una reserva.

2 Ejemplo

Como ejemplo, si tenemos 5 mesas en el restaurante y actualmente tenemos las siguientes reservas:

- Maria Reserva la mesa “2”
- Laia Reserva la mesa “5”
- Laia Reserva la mesa “1”

Es decir, dada esta situación de mesas:

COCINA	1	2	3	4	5	SALIDA
	x	x			x	

Es fácil y comprensible de ver que, la primera mesa libre es la “Mesa 3”.

3 Requerimientos

Dada esta problemática, se pide la realización de:

1. Un programa que adquiera los datos desde un fichero de texto y devuelva el valor óptimo de reservas.
2. Un informe (documento PDF) que contenga los siguientes aspectos:
 - a. Pseudocódigo del algoritmo principal en diseño iterativo y recursivo.
 - b. Estimación teórica y matemática de los costos de los algoritmos.
 - c. Costos experimentales de los programas.

4 Argumentos y parámetros



Para la realización de la actividad, lo que habrá que realizar es descomprimir e importar en PyCharm o similar el programa Python correspondiente. Dicho programa Python se podrá ejecutar de la siguiente forma:

```
$ python main.py <numero_reservas>
```

Complementariamente, el proyecto Python (fichero zip descomprimido) contiene una serie de test que se podrán ejecutar de la siguiente forma:

```
$ python reservesTest.py -v
```

Considerando estos dos aspectos, es importante tener en cuenta que, a la hora de realizar el ejercicio, es imprescindible cumplimentar la siguiente función que se encuentra dentro del fichero "main.py":

```
def bookerineManagement (reserves):  
  
    idTable = -1  
  
    #Your code here...  
  
    return idTable
```

5 Evaluación

La evaluación de la práctica se realizará sobre 10 puntos considerando la siguiente división de la puntuación:

- **Informe (3puntos)**¹. En el informe se considerarán los siguientes aspectos:
 - Análisis de costos teóricos (recursivo e iterativo) y empíricos (recursivo o iterativo) [1,25 puntos]
 - Descripción del pseudocódigo en iterativo y recursivo [1,25 punto]
 - Conversión de algoritmo iterativo a recursivo [0,5 puntos]
- **Codificación de los algoritmos (3 puntos)**. Codificación de los algoritmos en Python junto con una correcta documentación del mismo:
 - Codificación del algoritmo iterativo [1,5 puntos]
 - Codificación del algoritmo en recursivo [1,5 puntos]
- **Validación (4 puntos)**. Validación de los algoritmos en relación con el cumplimiento de los tests que adjuntan al proyecto en Python.

6 Envío de la práctica

La asignación se puede realizar por parejas y representa un 25% de la nota final. La práctica se debe presentar a través del campus virtual de la UdL en un fichero .zip o .rar que contenga:

¹ No se considerarán prácticas evaluables aquellas cuyo informe no sea un PDF y/o este no sea un documento correctamente formateado a través de un documento Word o similar. Es decir, no se admitirán prácticas como imágenes de un cuaderno o similar.



1. Un PDF con el informe donde se deben indicar los autores de la práctica.
2. Un archivo .zip o .tgz que contenga el código correspondiente desarrollado.