



---

IIC2115 – Programación como Herramienta para la Ingeniería (I/2021)

## Taller 1b

### Objetivos

- Utilizar estructuras de datos básicas.

### Entrega

- **Lenguaje a utilizar:** Python 3.6 o superior
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **T1b**.
- **Entrega:** lunes 29 de marzo a las 16:50 hrs.
- **Formato de entrega:** archivos python notebook (**T1b.ipynb**) y python simple (**T1b.py**) con la solución a lo solicitado en el enunciado. Ambos archivos deben estar ubicados en la carpeta **T1b**. No se debe subir ningún otro archivo a la carpeta. En el archivo python notebook, utilice múltiples celdas de texto y código para facilitar la revisión de su programa. El archivo python simple debe ser exportado a partir del python notebook.
- **NO SE ADMITEN ENTREGAS POR CORREO U OTROS MEDIOS NI ENTREGAS FUERA DE PLAZO**
- **Entregas con errores de sintaxis y/o que generen excepciones serán calificadas con nota 1.0.**

### Introducción

Con el fin de ejercitar el uso de estructuras de datos, en este taller deberá desarrollar 2 ejercicios. El primero será guiado a través de instrucciones en el enunciado, mientras que en el segundo el desarrollo será por su cuenta.

## Ejercicio 1: conteo de ocurrencias

Utilizando estructuras de datos, construya un programa que cuente las ocurrencias de cada palabra distinta en un texto. Para esto, siga las instrucciones descritas a continuación:

1. El texto será entregado como una lista de palabras (*strings*). Considere que puede haber palabras repetidas en este.
2. Cree un diccionario vacío, donde se almacenaran pares (llave, valor), donde la llave será una palabra y el valor la cantidad de veces que esta ha aparecido.
3. Itere sobre la lista de palabras, **verificando para cada una si esta se encuentra ya almacenada en el diccionario**. Si no lo está, cree un par (llave, valor) para la palabra, fijando el número de ocurrencias como cero. Si la palabra ya está almacenada en el diccionario, incremente en 1 su número de ocurrencias.
4. Una vez concluido el proceso, obtenga los ítems del diccionario y transfórmelos en una lista de tuplas, donde cada una tiene el formato (palabra, número ocurrencias).
5. Empaque todo esto una función que reciba como argumento una lista de palabras (*strings*) y que retorne una lista de tuplas, donde cada una contiene en su primer elemento una palabra de la lista y en el segundo la cantidad de veces que esa ocurre en la lista. Un ejemplo de ejecución del algoritmo es el siguiente:

### Código

```
palabras = ['avión', 'perro', 'gato', 'avión', 'edificio', 'gato']
ocurrencias = calcular_ocurrencias(palabras)
print(ocurrencias)
```

### Salida

```
[('avión', 2), ('perro', 1), ('gato', 2), ('edificio', 1)]
```

## Ejercicio 2: suma de cuadrados

Dada una lista  $L$  de números naturales, encuentre todos los tríos  $(L[i], L[j], L[k])$  con  $i \neq j \neq k$ , tales que  $L[i]^2 + L[j]^2 = L[k]^2$ . La solución entregada no puede estar basada en fuerza bruta, *i.e.*, no se puede solucionar el problema probando cada uno de los posibles tríos  $(i, j, k)$ .

### Código

```
numeros = [2,3,4,6,7,12,13,15,5,17,14,22]
trios = suma_cuadrados(numeros)
print(trios)
```

### Salida

```
[(3,4,5),(5,12,13)]
```

## Objetivo parcial de participación

Para verificar la participación durante la clase, terminar el ejercicio 1.

## Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

*“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”*

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en

material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por “copia” o “plagio” se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.