



## MiniProyecto 3



### Recetas

El objetivo de este proyecto es que puedas poner en práctica la mayor parte de lo aprendido sobre programación en Python, específicamente lo que se refiere a diccionarios, listas, funciones y archivos.

#### Instrucciones para realizar la actividad:

Existen dos archivos con información para un restaurant: `ingredientes.txt` y `recetas.csv`. En el primero se guarda la información de los ingredientes que se usan en las distintas recetas junto a su stock. El segundo archivo guarda la información de los ingredientes que van en cada una de las recetas.



A continuación, un ejemplo de cada uno de estos dos archivos:

#### **ingredientes.txt**

```
Tomate 5
Lechuga 2
Hamburguesa 10
Carne 1
Espárragos 8
Pan 10
Papa 5
Cebolla 12
```

#### **recetas.csv**

```
HamburguesaCasera,Hamburguesa,Tomate,Pan
PastelDeCarne,Carne,Papa,Cebolla
EnsaladaEspecial,Lechuga,Espárragos,Tomate
```

Para simplificar el problema puedes suponer que los datos en el archivo siempre vienen correctos en la forma indicada. Es decir en el archivo de ingredientes el nombre y el stock separados por espacios y en el archivo de recetas el nombre de la receta y luego todos los ingredientes separados por comas.

Tu misión es escribir un programa capaz de responder a las instrucciones del usuario en la siguiente forma. Hay tres comandos: PREPARAR, REPONER, y STOP. Cualquier otra palabra inicial que no sea una de ellas se ignora.

PREPARAR viene seguida del nombre de una de las recetas del archivo.

REPONER viene seguida de una lista de ingredientes separados por espacios. Pueden haber ingredientes que aparezcan más de una vez

STOP no es acompañada por nada.

Si la orden es PREPARAR debe procederse a verificar si la receta es válida y si tenemos los ingredientes. En ese caso se reduce en una unidad cada uno de los ingredientes que la receta indicada incluye. Si la receta no existe en el archivo de recetas (por ejemplo, Omelette) se imprime el siguiente mensaje:

\*\*\* Lo sentimos pero no preparamos Omelette \*\*\*

Si la receta es válida (por ejemplo, PastelDeCarne), pero hay algún ingrediente del cual no tenemos stock (por ejemplo stock = 0 para Carne) se imprime el siguiente mensaje:

\*\*\* No se puede hacer PastelDeCarne porque falta Carne \*\*\*

Si la orden es REPONER debe procederse a aumentar en una unidad el stock de cada uno de los ingredientes indicados (si aparece varias veces se agrega por cada aparición. Si algún producto de los ingresados no existe en el archivo simplemente se ignora.

Cada vez que se pide PREPARAR o REPONER, luego de realizar la acción debe desplegarse una lista de los ingredientes con sus respectivos stocks actualizados en la forma:

```
Stock actual de ingredientes disponibles
Tomate 5
Lechuga 2
Hamburguesa 10
Carne 2
Espárragos 8
Pan 11
Papa 4
Cebolla 11
```

Si la orden es STOP el programa se detiene

Mientras no se ingrese la orden de STOP el programa vuelve a pedir una nueva orden del usuario.

## Ejemplo de Uso

Ingresa la receta que quieres o REPONER: PREPARAR PastelDeCarne

Stock actual de ingredientes disponibles

Tomate 5

Lechuga 2

Hamburguesa 10

Carne 0

Espárragos 8

Pan 10

Papa 4

Cebolla 11

Ingresa la receta que quieres o REPONER: PREPARAR PastelDeCarne

\*\*\* No se puede hacer PastelDeCarne porque falta Carne \*\*\*

Stock actual de ingredientes disponibles

Tomate 5

Lechuga 2

Hamburguesa 10

Carne 0

Espárragos 8

Pan 10

Papa 4

Cebolla 11

Ingresa la receta que quieres o REPONER: REPONER Leche Carne Huevos Carne Pan

Stock actual de ingredientes disponibles

Tomate 5

Lechuga 2

Hamburguesa 10

Carne 2

Espárragos 8

Pan 11

Papa 4

Cebolla 11

Stock actual de ingredientes disponibles

Tomate 5

Lechuga 2

Hamburguesa 10

Carne 2

Espárragos 8

Pan 11

Papa 4

Cebolla 11

Ingresa la receta que quieres o REPONER: PREPARAR Omelette

\*\*\* Lo sentimos pero no preparamos Omelette \*\*\*

Stock actual de ingredientes disponibles

Tomate 5

Lechuga 2

Hamburguesa 10

Carne 2

Espárragos 8

Pan 11

Papa 4

Cebolla 11

Ingresa la receta que quieres o REPONER: PREPARAR EnsaladaEspecial

Stock actual de ingredientes disponibles

Tomate 4

Lechuga 1

Hamburguesa 10

Carne 2

Espárragos 7

Pan 11

Papa 4

```

Cebolla 11
Ingresa la receta que quieres o REPONER: PREPARAR EnsaladaEspecial
Stock actual de ingredientes disponibles
Tomate 3
Lechuga 0
Hamburguesa 10
Carne 2
Espárragos 6
Pan 11
Papa 4
Cebolla 11
Ingresa la receta que quieres o REPONER: PREPARAR EnsaladaEspecial
*** No se puede hacer EnsaladaEspecial porque falta Lechuga ***
Stock actual de ingredientes disponibles
Tomate 3
Lechuga 0
Hamburguesa 10
Carne 2
Espárragos 6
Pan 11
Papa 4
Cebolla 11
Ingresa la receta que quieres o REPONER: STOP

Process finished with exit code 0

```

## Ayuda para construir el programa

### Construcción de Diccionarios

Dado que necesitaremos constantemente consultar para un ingrediente dado si existe o si tenemos stock es buena idea construir un diccionario en que la llave sea el nombre del ingrediente:

```
diccionario_ingredientes = {'Tomate': 3, 'Lechuga': 0, ...}
```

También necesitamos acceder a una receta dada por su nombre por lo que resulta conveniente construir otro diccionario:

```

diccionario_recetas = {
    'PastelDeCarne': ['Carne', 'Papa', 'Cebolla'],
    'EnsaladaEspecial': ['Lechuga', 'Espárragos', 'Tomate'],
    ...
}

```

Observa que los valores en el segundo diccionario son listas (con los ingredientes de la receta)

Al iniciar el programa lo primero es leer el archivo ingredientes.txt y con esa información construir el diccionario de ingredientes. Luego leer el archivo recetas.csv y con esa información construir el diccionario de recetas.

## Funciones

Se recomienda escribir las siguientes funciones:

`cargar_diccionario_ingredientes()` - lee el archivo `ingredientes.txt` y con esa información carga el diccionario de ingredientes a ser usado mas adelante

`cargar_diccionario_recetas()` - lee el archivo `recetas.csv` y con esa información carga el diccionario de recetas a ser usado mas adelante

`printStocks()` - A partir del diccionario de ingredientes genera el listado que debe imprimirse después de cada operación de preparación o reposición

`reponerIngredientes(lista_ingredientes)` - se le pasa la lista de ingredientes a reponer y le suma uno al stock de cada uno de los que aparecen en el diccionario

`prepararReceta(receta)` - se le pasa el nombre de una receta con lo cual extrae desde el diccionario de recetas la lista de ingredientes para actualizar con ello el diccionario de ingredientes (ojo con los casos especiales)

Se recomienda probar cada una de estas funciones en forma separada hasta asegurar que funcionan bien. Una vez que tienes escritas estas funciones lo que queda es muy simple y corresponde a un simple ciclo en que se pide un comando y luego de leerlo se llama a las funciones correspondientes.

## Algunas funciones disponibles en Python que te pueden ayudar

`f.readline()` - lee la siguiente línea del archivo asociado al descriptor `f` (se obtiene al abrirlo). Al final se lee un string vacío (`""`)

`s.rsplit()` - genera una lista con cada una de las palabras de un string `s` separadas por espacios. Si el parámetro es `','` considera la separación por comas en lugar de blancos

`lista1.append(lista2)` - los elementos de `lista2` se agregan al final de los de `lista1`

`len(lista1)` - entrega el número de elementos en `lista1`

`linea.rstrip("\n")` - elimina el carácter final de `newline`

`lista1.pop(0)` - devuelve el primer elemento de `lista1` (índice 0) y retira de `lista1` dicho elemento

`x in diccionario` - devuelve `True` si la llave `x` se encuentra en el diccionario y `False` en caso contrario