

Plan Formativo Full Stack Python	
Módulo	Programación Avanzada en Python
Tema	Planta productora de flores.
Nivel de Dificultad	Alta
Duración	3.5 horas
Ejecución	Grupal (3 - 4 personas)
Intención del aprendizaje o aprendizaje esperado:	
<ul style="list-style-type: none"> • Aplicación de conocimientos de python avanzado. • Demostración de capacidad de realizar programación orientada a objetos (Clases, Herencia, Polimorfismos, Encapsulamiento, etc.) • Elaboración de diagrama de clases. • Manejo de errores. • Manejo de archivos para procesamiento de datos y almacenamiento. • Implementación de programa de funcionamiento dinámico con flujo infinito. 	
Planteamiento del Problema:	
<p>FlorLinda Valdivia posee una planta de producción de ramos de flores. La empresa compra flores de diferentes especies y tamaños para completar sus pedidos.</p> <p>Los ramos producidos son de diferentes diseños y tamaños, según especificaciones de diseño que indican cantidad de flores y de qué tipo.</p> <p>Las flores llegan una a una a la bodega y se almacenan hasta que hay cantidad suficiente para producir un ramo.</p> <p>Se solicita crear un programa que procese el flujo de flores de llegada y obtenga el flujo de ramos como su resultado.</p> <p>Debe crear una interfaz por línea de comandos programada en Python3.</p> <p>Especificaciones de formato de entrada / salida:</p> <ul style="list-style-type: none"> • especie_flor: Letra única minúscula: a-z. • tamano_flor: Letra única mayúscula: L (Grande), S (Pequeña). • flor: Un par especie_flor y tamano_flor: por ejemplo rL. • nombre_ramo: Letra única mayúscula: A-Z. • tamano_ramo: Letra única mayúscula: L (Grande), S (Pequeña). • diseno_ramo: Línea única de caracteres con el formato: 	

```
<nombre_ramo><tamano_ramo><cantidad_flor-1><especie_flor-1> ...  
    <cantidad_flor-N><especie_flor-N><cantidad_flores_ramo>
```

Por ejemplo: AL8d10r5t30

- **ramo:** Línea única de caracteres con el formato:

```
<nombre_ramo><tamano_ramo><cantidad_flor-1><flower 1 especie>...  
    <cantidad_flor-N><especie_flor-N>
```

Por ejemplo: AL8d10r5t7z

- Los formatos de **diseno_ramo** y **ramo** incluyen un tamaño de ramo pero no tamaños de flores. Esto se debe a que solo se hacen ramos grandes de flores grandes, y los ramos pequeños solo están hechos de flores pequeñas.
- **especies_flor** se enumera en orden alfabético y solo aparecen una vez tanto en los diseños de ramo como en los ramos de flores.
- **cantidad_flor-i** es siempre superior a 0 tanto en **diseno_ramo** como en **ramo**.
- **cantidad_flores_ramo** en **diseno_ramo** puede ser mayor que la suma de las cantidades de flores, lo que permite un espacio extra en los ramos que puede consistir en cualquier tipo de flores.
- **ramo** no tiene la cantidad total de flores especificado, pero la suma de las cantidades de flores debe ser igual a la cantidad total de flores en el **diseno_ramo** correspondiente.
- El flujo de entrada de flores posee la estructura:

```
diseno_ramo-1  
diseno_ramo-2  
<línea vacía>  
flor-1  
flor-2  
flor-3  
...
```

Por ejemplo:

```
AS3a4b6k20  
AL8d10r5t30
```

```
aS  
aS  
bL  
rL  
tS  
...
```

- La salida debe ser un **ramo** cada vez que se pueda crear uno a partir de las flores disponibles de acuerdo con uno de los elementos **diseno_ramo** proporcionados:

Por ejemplo:

AL8d10r5t7z
AS10a4b6k
...

Se solicita:

- Definir los roles del grupo de desarrollo y esquema de ramas para trabajo en base a repositorio Git/GitHub.
- El diagrama de clases de la solución propuesta (Se sugiere tener una discusión grupal hasta concluir el mejor diseño).
- Debe procurar utilizar de la manera más óptima posible el uso de Clases, herencias, composición, encapsulamiento y todos los conceptos estudiados en relación a programación orientada a objetos.
- Se sugiere que luego de un avance en la implementación de su diseño, se vuelva a discutir el diagrama de clases para intercambiar opiniones sobre eventuales optimizaciones de éste.
- Debe entregar un programa funcional capaz de ser probado por un usuario de perfil no técnico.
- El código final debe quedar en un repositorio GitHub, donde debe estar visible las ramas de desarrollo de los distintos integrantes con su aporte, además de las ramas master, desarrollo, pruebas u otras que el grupo haya decidido utilizar.

Recursos Bibliográficos :

[1] Modulo 3: Programación Avanzada en Python