```python
#i. A una variable se le asigne un mensaje motivador que incluya los
nombres de todos los
#integrantes. ¿Qué tipo de dato puede ser?

mensaje = "Compra con nosotros, empresa desarrollada por Patricio,
Francisco, Ricardo y Javier"
#print(mensaje)

#ii. Se asegure que todos su caracteres estén en mayúscula.

mensaje=mensaje.upper()
#print(mensaje)

#iii. Elabore una lista con cada palabra del string.

mensaje=mensaje.replace(",","")
lista_mensaje=mensaje.split()
#print(lista_mensaje)

#iv. Cada integrante del grupo debe reconocer en qué índice está su
nombre.


for integrante in ["Ricardo","patricio","FRANcIsCO","JAviER"]:
    print(f"index {integrante.upper()} es
{lista_mensaje.index(integrante.upper())}")

#v. Indique cuántas palabras tenía el string.

print(f"la frase tiene {len(lista_mensaje)} palabras")

#vi. Imprima una tupla con todas las palabras del string.

tupla_mensaje=tuple(lista_mensaje)
print(tupla_mensaje)

#vii. ¿Con qué función podrían obtener el mensaje por teclado al ejecutar
el programa? Implementarlo!.

while True:
    respuesta=input("quieres un mensaje motivador? si/no").lower()
    if respuesta=="si":
        print(mensaje)
        break
    if respuesta=="no":
        print("que pena")
```

```
        break
    else:
        print("ingrese una respuesta valida (si/no)")
```

- Discutan ¿Qué es un dato booleano? ¿Qué utilidad puede tener para el desarrollo de un

programa?

Es un valor lógico: verdadero o falso. Sirve para tomar decisiones lógicas durante la ejecución del código.

- Investigar qué significa que python sea un lenguaje de tipado dinámico.

Puede definirse variables sin necesidad de declararlas ni determinar el tipo de dato asociado a priori.

- Investigar y documentar sobre la creación de Módulos en Python.

Un archivo.py puede contener métodos y/o funciones, útiles para un problema, y puede importarse.

- Investigar y documentar sobre la creación de Paquetes en Python.

No tiene mucho misterio, basta con definir funciones, documentarlas bien y guardar todo en un archivo que no será usado para ejecutar tales funciones. Se importa el archivo como un módulo y se ejecutan las funciones referenciando al modulo. Un ejemplo son las API.

- Investigar e implementar el uso del archivo __init__.py

__init__.py permite ordenar un directorio que tiene muchos módulos, de manera tal que se puede importar el directorio y cada modulo con una sintaxis tal como

```
import directorio

objeto1 = directorio.funcion1()
objeto2 = directorio.funcion2()
objeto3 = directorio.funcion3()
```

en vez de usar

```
from modulo import funcion1
from modulo import funcion2
from modulo import funcion3
```

uno a uno. Entonces el __init__.py en el directorio debe tener estos from…