



Markdown



```
print(pals_ordenadas)
```

```
[('me', 4), ('tengo', 1), ('en', 3), ('hace', 2)]
[('me', 4), ('en', 3), ('hace', 2), ('tengo', 1)]
['me', 'en', 'hace', 'tengo']
```

Pongamos lo anterior todo junto:

```
In [7]: ▶ def palabras_frecuentes(palabras):
        lista = [(w, palabras.count(w)) for w in palabras]
        lista_sin_dups = []
        for wf in lista:
            if wf not in lista_sin_dups:
                lista_sin_dups.append(wf)
        pal_frecs_ordenadas = sorted(pal_frecs, key = lambda wf: wf[1], reverse=True)
        return [w for (w, f) in pal_frecs_ordenadas]

        lista_inicial = ['me', 'me', 'tengo', 'me', 'en', 'me', 'en', 'hace', 'en', 'hace']

        print(palabras_frecuentes(lista_inicial))

        ['me', 'en', 'hace', 'tengo']
```

Ahora un diccionario de sugerencias dará, para cada palabra, la lista de posibles continuaciones, ordenadas de mayor a menor frecuencia.

Dejo esto pendiente como un pequeño desafío para ti... :-)