# Lista 05: Expresiones Regulares

La lista muestra un buen manejo de las expresiones regulares y su implementación para resolver problemas comunes en la manipulación de cadenas. El código es funcional y refleja un esfuerzo por cumplir con los requisitos de los ejercicios. Muestra una comprensión sólida de las expresiones regulares y su aplicación en Python. Las soluciones son funcionales y cumplen con los objetivos planteados. Implementar las mejoras sugeridas podría elevar aún más la calidad de su trabajo. ¡Buen trabajo!

**Cosas positivas:**

* Utiliza patrones adecuados para la extracción y manipulación de datos con expresiones regulares.
* Integra las funciones re.findall, re.sub y re.match de manera efectiva.
* Presenta soluciones claras y bien estructuradas.

**Posibles mejoras:**

* Incluir comentarios más detallados para explicar las expresiones regulares, facilitando la comprensión del código.
* Refinar las expresiones regulares para hacerlas más robustas y generales.
* Evitar redundancia en las expresiones regulares y simplificar el código cuando sea posible.

### Ejercicio 1

**Estado del ejercicio:** Correcto.  
**Errores encontrados:** Ninguno.  
**Código optimizado:** No es necesario.  
**Comentario:** La expresión regular utilizada (\W+) cumple con el objetivo de separar el texto en palabras ignorando los caracteres no alfanuméricos. Es una solución eficiente y bien planteada.

### Ejercicio 2

**Estado del ejercicio:** Correcto.  
**Errores encontrados:** Ninguno.  
**Código optimizado:** No es necesario.  
**Comentario:** La expresión regular \d+ identifica correctamente los años en el texto. Sin embargo, se podría especificar un rango de longitud (\b\d{4}\b) si solo se buscan números de 4 dígitos.

### Ejercicio 3

**Estado del ejercicio:** Correcto.  
**Errores encontrados:** Ninguno.  
**Código optimizado:** No es necesario.  
**Comentario:** El patrón (\W)\1+ es eficiente para reducir múltiples símbolos consecutivos a uno solo. La implementación es clara y funcional.

### Ejercicio 4

**Estado del ejercicio:** Correcto.  
**Errores encontrados:** Ninguno.  
**Código optimizado:** No es necesario.  
**Comentario:** La expresión regular para detectar palabras repetidas es adecuada y funcional. Sin embargo, añadir ejemplos adicionales en los comentarios ayudaría a ilustrar mejor su uso.

### Ejercicio 5

**Estado del ejercicio:** Correcto.  
**Errores encontrados:** Ninguno.  
**Código optimizado:** No es necesario.  
**Comentario:** El uso de grupos nombrados en la expresión regular es excelente para organizar y extraer tanto correos electrónicos como importes. Es una solución profesional y bien estructurada.

### Ejercicio 6

**Estado del ejercicio:** Correcto.  
**Errores encontrados:** Ninguno.  
**Código optimizado:** No es necesario.  
**Comentario:** La expresión regular captura los diferentes formatos de números de teléfono y los transforma en un formato uniforme de manera eficiente. Es una solución bien diseñada.

### Ejercicio 7

**Estado del ejercicio:** Correcto.  
**Errores encontrados:** Ninguno.  
**Código optimizado:** No es necesario.  
**Comentario:** La validación de correos electrónicos con la expresión regular es robusta y cumple con los requisitos básicos. Sin embargo, podría beneficiarse de una validación más avanzada si se desea cubrir casos adicionales como dominios más complejos.