

## 1. Anexos

### 1.1. Anexo I. Relación del trabajo con los Objetivos de Desarrollo Sostenible

El presente trabajo se relaciona con el ODS 3 Salud y bienestar ya que la investigación en cómo el hígado se regenera puede llevar a la aparición de nuevos tratamientos contra enfermedades hepáticas y la mejora de la calidad de vida de los afectados.

<b>Objetivos de Desarrollo Sostenible</b>	<b>Alto</b>	<b>Medio</b>	<b>Bajo</b>	<b>No procede</b>
ODS 1. Fin de la pobreza				X
ODS 2. Hambre cero				X
ODS 3. Salud y bienestar	X			
ODS 4. Educación de calidad				X
ODS 5. Igualdad de género				X
ODS 6. Agua limpia y saneamiento				X
ODS 7. Energía asequible y no contaminante				X
ODS 8. Trabajo decente y crecimiento económico				X
ODS 9. Industria, innovación e infraestructuras				X
ODS 10. Reducción de las desigualdades				X
ODS 11. Ciudades y comunidades sostenibles				X
ODS 12. Producción y consumo responsables				X
ODS 13. Acción por el clima				X
ODS 14. Vida submarina				X
ODS 15. Vida en ecosistemas terrestres				X
ODS 16. Paz, justicia e instituciones sólidas				X
ODS 17. Alianzas para lograr objetivos				X

## 1.2. Anexo II. Script en Python para el remuestreo.

```
import random

# Definir la lista de elementos y sus frecuencias
elementos = ["Fb", "Hp", "LEC", "IC", "PC"]
frecuencias = [54, 68, 11, 21, 3]

# Definir la combinación objetivo
combinacion_objetivo = ["Hp", "Hp"]

# Calcular la probabilidad teórica de la combinación objetivo
probabilidad_teorica = 1
total_frecuencias = sum(frecuencias)

for elemento in combinacion_objetivo:
    frecuencia_elemento = frecuencias[elementos.index(elemento)]
    probabilidad_elemento = frecuencia_elemento / total_frecuencias
    probabilidad_teorica *= probabilidad_elemento

# Definir el número de iteraciones de Monte Carlo
num_iteraciones = 100000

# Realizar el análisis de Monte Carlo
ocurrencias = 0

for _ in range(num_iteraciones):
    # Generar una combinación aleatoria
    combinacion_aleatoria = random.choices(elementos, weights=frecuencias,
k=len(combinacion_objetivo))

    # Comprobar si la combinación aleatoria coincide con la combinación
objetivo
    if combinacion_aleatoria == combinacion_objetivo:
        ocurrencias += 1

# Calcular la probabilidad observada
probabilidad_observada = ocurrencias / num_iteraciones

# Imprimir el resultado
print("Probabilidad observada:", probabilidad_observada)
```