



**UNIVERSIDAD POLITÉCNICA DE TLAXCALA REGIÓN  
PONIENTE**

**REPORTE DE PROBLEMA DEL COCO**

**SISTEMAS INTELIGENTES**

**PROFA: ING. VANESA TENOPALA ZAVALA**

**INTEGRANTES DEL EQUIPO:**

**INGRID MAGALI CRUZ MALDONADO**

**ANTONIO OLVERA ESPINOZA**

**DIANA LAURA PEÑA ROJAS**

**RUBI JUAREZ GALLARDO**

**LEO XELHUANTZI LIRA**

**ING. SISTEMAS COMPUTACIONALES**

**GRADO: 8to**

**GRUPO: "A"**

**ENERO-ABRIL 2024**



## RESUMEN

El problema de los cocos y los naufragos se empezó con un primer código para implementar un algoritmo iterativo que prueba diferentes cantidades de cocos iniciales hasta encontrar la cantidad mínima que satisface las condiciones de división establecidas en el problema.

En caso del código HTML proporciona una interfaz para calcular y visualizar la distribución de cocos entre cinco naufragos y un mono. Al presionar un botón, se ejecuta una función en JavaScript que realiza los cálculos necesarios y muestra los resultados. El programa simula una situación de repartición equitativa de recursos en un escenario de supervivencia, donde se explora cómo distribuir eficientemente los cocos entre los naufragos y el mono para garantizar la justa distribución de recursos.

## INTRODUCCIÓN

El problema de los cocos y los monos es un enigma matemático clásico que desafía a encontrar el número mínimo de cocos iniciales que deben recolectarse para que una división específica de cocos entre un grupo de monos deje un residuo particular.

En este contexto, el código que aborda este problema mediante un enfoque algorítmico, utilizando Python como lenguaje de programación y un código HTML y JavaScript que simula la distribución equitativa de cocos entre cinco naufragos y un mono.

## DESARROLLO

5 hombres y un mono naufragan en una isla desierta. Los naufragos pasan todo el primer día recogiendo cocos, por la noche uno de ellos se despierta y decide separar su parte divide los cocos en 5 montones iguales y como sobra un coco se lo da a el mono y oculta su parte. Poco más tarde un segundo naufrago despierta y hace lo mismo al dividir los cocos en 5 montones también sobra un coco y de igual forma se lo da a el mono y oculta su parte, cada uno de los naufragos se levanta y hace lo mismo que los anteriores, por la mañana agrupan los cocos en 5 montones iguales y esta vez no sobra ningún coco

- 1.-Cuántos cocos eran inicialmente?
- 2.-Cuántos se quedó el mono?
- 3.-Cuántos se quedó cada naufrago?

```
def cocos():  
    cocos_totales = 1  
  
    while True:  
        cocos_en_montones = cocos_totales  
        sobran_cocos = 0  
  
        for _ in range(5):  
            if cocos_en_montones % 5 != 1:  
                sobran_cocos = 1  
                break
```



```
cocos_en_montones = (cocos_en_montones - 1) // 5 * 4

if sobran_cocos == 0:
    return cocos_totales

cocos_totales += 1

cocos_iniciales = cocos()

print("Cocos inicialmente:", cocos_iniciales)
```

```
<script>
function repartirCocos() {
    const cocosPorNaufrago = [];
    let cocosRestantes = totalCocos;

    for (let i = 0; i < 5; i++) {
        const cocosParaMono = Math.floor(cocosRestantes / 5);
        cocosPorNaufrago.push(cocosParaMono);
        cocosRestantes -= cocosParaMono;
        cocosRestantes--;
    }

    let cocosParaMonoPorLaMañana = cocosRestantes;
    let cocosPorNaufragoPorLaMañana = Math.floor((totalCocos - 5) / 5);

    const resultsList = document.getElementById('results-list');
    resultsList.innerHTML = '';

    let totalCocosNaufragos = 0;

    for (let i = 0; i < 5; i++) {
        const li = document.createElement('li');
        li.innerHTML = `<span class="circle blue">👤</span> Naufrago ${i + 1}:
${cocosPorNaufrago[i]} cocos escondidos.<br>Y despues de la reparticion final
tiene : ${cocosPorNaufrago[i]+204} cocos <br><br> `;
        resultsList.appendChild(li);

        totalCocosNaufragos += cocosPorNaufrago[i];
    }

    const monoLi = document.createElement('li');
    monoLi.innerHTML = `<span class="circle red">👤</span> Mono: 5 cocos`;
    resultsList.appendChild(monoLi);

    const liTotalMañana = document.createElement('li');
```



```
liTotalMañana.innerHTML = `<span class="circle brown"></span> A cada  
naufrago en la reparticion final le tocaron: ${cocosRestantes/5} cocos`;  
resultsList.appendChild(liTotalMañana);  
  
const resultsDiv = document.getElementById('results');  
resultsDiv.style.display = 'block';  
}  
</script>
```

## EJECUCIÓN

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
PS C:\Users\66422> & C:/Python311/python.exe c:/Users/66422/Untitled-1.py  
Cocos inicialmente: 3121  
PS C:\Users\66422>
```

Cantidad de cocos recolectados:



3121 cocos

Reparticion de cocos

3121 cocos

Reparticion de cocos

**Resultado:**

-  Naufrago 1: 624 cocos escondidos.  
Y despues de la reparticion final tiene : 828 cocos
-  Naufrago 2: 499 cocos escondidos.  
Y despues de la reparticion final tiene : 703 cocos
-  Naufrago 3: 399 cocos escondidos.  
Y despues de la reparticion final tiene : 603 cocos
-  Naufrago 4: 319 cocos escondidos.  
Y despues de la reparticion final tiene : 523 cocos
-  Naufrago 5: 255 cocos escondidos.  
Y despues de la reparticion final tiene : 459 cocos
-  Mono: 5 cocos
-  A cada naufrago en la reparticion final le tocaron: 204 cocos

## CONCLUSIÓN

Este programa no solo proporciona entretenimiento, sino que también ilustra conceptos importantes como la equidad en la distribución de recursos. Al simular una situación de supervivencia donde los recursos son limitados, el programa destaca la importancia de garantizar una distribución justa y equitativa para todos los involucrados. Además, al ofrecer una experiencia interactiva y visualmente atractiva, el programa ayuda a los usuarios a comprender mejor cómo se pueden abordar estos problemas en la vida real.