

# Tercer Proyecto Programado:

## Decodificador Gráfico de Mensajes

**Curso:** Taller de programación

**Fecha Inicio:** martes 27 de mayo 2025 12:30 pm

**Fecha de entrega:** martes 24 de junio 2025, 08:00 am

**Valor Porcentual:** 30%

---

## Descripción General

Crear una aplicación en Python que permita **cifrar mensajes y descifrarlos** por medio de **archivos binarios y con la implementación de un árbol binario, reconstruya visualmente el árbol y muestre gráficamente el proceso de decodificación paso a paso**, ilustrando cómo se recorre el árbol para reconstruir el mensaje.

## Funciones principales

1. Recibir mensaje y cifrarlo en un archivo binario.
2. Leer un archivo y descifrarlo.
3. Graficar árbol y desplegar de forma animada el descifrado del mensaje.

## Estructura del archivo .bin

Sección	Descripción
<b>Cantidad de caracteres codificados</b>	4 bytes: representa la cantidad de caracteres distintos codificados
<b>Caracteres y sus repeticiones</b>	Secuencia de: 1 byte para el carácter y 2 bytes para



	sus repeticiones.
<b>Bits descartados</b>	1 Byte: Cantidad de bits descartados en el último byte escrito del mensaje codificado
<b>Mensaje codificado</b>	Secuencia de bits que representan el mensaje codificado según el árbol

## Ejemplo de codificación (Preorden Serializado)

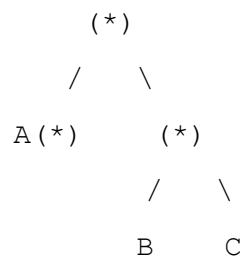
Mensaje a codificar:

"ABAC"

Vamos a asumir esta frecuencia:

- A: 2 veces
- B: 1 vez
- C: 1 vez

Entonces un árbol de Huffman posible es:



Codificación:

- A → 0
- B → 10
- C → 11

Codificación del mensaje "ABAC":

Letra    Código

A	0
B	10
A	0
C	11

Resultado codificado en bits:

0 10 0 11 → 010011

Estructura del archivo esperada para el ejemplo anterior:

- Primeros 4 Bytes: representación en 4 bytes del número 3, que representan los 3 caracteres de la lista de frecuencias.



- 00000011
- Secuencia de caracteres y su frecuencia: 9 bytes para representar los caracteres y sus frecuencias (A → 2 , B → 1 , C → 1)
  - A: 01000001 , 2: 00000000 00000010
  - B: 01000001 , 1: 00000000 00000001
  - C: 01000001 , 1: 00000000 00000001
- 1 Byte que representa los bits de descarte: en este caso representando el número 2 por que el código generado solo tienen 6 bits.
  - 00000010
- Secuencia de la codificación en grupos de bytes hasta finalizar el archivo
  - 00010011

Agrupando los bits:

- 01000001 → 0x41
- 00000000 → 0x00
- 00000010 → 0x02
- 01000001 → 0x41
- 00000000 → 0x00
- 00000001 → 0x01
- 01000001 → 0x41
- 00000000 → 0x00
- 00000001 → 0x01
- 00000010 → 0x02
- 00010011 → 0x13

Paso 2: Resultado en hexadecimal

41 00 02 41 00 01 41 00 01 02 13

## ¿Cómo codificar un archivo .bin?

1. Construir el árbol binario (por ejemplo, de Huffman).
2. Generar los caracteres y su frecuencia
3. Generar la tabla de codificación: cada carácter → camino de bits.
4. Codificar el mensaje como una secuencia de bits.

## ¿Cómo decodificar?

1. Reconstruir el árbol desde los caracteres y su frecuencia
2. Leer los bits codificados.



3. Recorrer el árbol bit a bit: 0 → izquierda, 1 → derecha.
4. Mostrar el carácter cuando se llega a una hoja.
5. Repetir hasta terminar.

## Visualización gráfica

Herramientas recomendadas:

- Pygame, Tkinter o PyQt para interfaz GUI.

Comportamiento esperado:

1. Al cargar el archivo .bin, se visualiza el árbol.
2. Cada bit se recorre con una animación:
3. Se resalta el nodo visitado (por ejemplo, cambiando color).
4. Cuando se llega a una hoja, se muestra el carácter decodificado en una barra inferior.
5. El mensaje final se reconstruye y se muestra dinámicamente.

## Ejemplo visual paso a paso

1. Archivo cargado:
2. Árbol aparece dibujado.
3. Bit leído: 0
4. Se resalta la rama izquierda.
5. Bit siguiente: 1
6. Se resalta rama derecha del nodo anterior.
7. Llegó a hoja: muestra "E", agrega a la salida.
8. Continúa...

## Rúbrica adaptada (añadiendo visualización)

Categoría	Descripción	Puntos Máximos
<b>1. Codificación del Mensaje</b>	Implementa correctamente la lectura del mensaje y su codificación en bits según un árbol binario.	<b>20 pts</b>
<b>2. Estructura del Archivo .bin</b>	El archivo binario generado sigue el formato especificado: cabecera, frecuencias, bits descartados y codificación.	<b>20 pts</b>





<b>3. Decodificación del Archivo .bin</b>	Reconstruye correctamente el árbol y decodifica el mensaje desde el archivo.	<b>20 pts</b>
<b>4. Visualización del Árbol Binario</b>	Dibuja correctamente el árbol binario en una GUI.	<b>15 pts</b>
<b>5. Animación del Proceso de Decodificación</b>	Ilustra visualmente y paso a paso cómo se recorre el árbol para reconstruir el mensaje.	<b>15 pts</b>
<b>6. Interfaz de Usuario (GUI)</b>	Interfaz clara, funcional y comprensible.	<b>10 pts</b>
<b>7. Estructura y Calidad del Código</b>	Código bien estructurado, modular y con buenas prácticas.	<b>10 pts</b>

## Aspectos Administrativos

1. El proyecto puede ser desarrollado en grupos de 2 personas, cualquier intento de copia o plagio será castigado con la anulación del proyecto.
2. Los fuentes del proyecto junto con la documentación deberán ser entregados a más tardar el día martes 13 de junio 2023, 08:00 am, por medio del la plataforma Moodle.
3. Retrasos en la fecha y hora de entrega se castigará en la nota final del proyecto. Cada hora de retraso disminuirá en un 10% la nota final del proyecto.
4. Debe hacerse uso de algún sistema de control de versiones que los estudiantes definan y asignar permisos de acceso al docente para su respectivo seguimiento.
5. Cada fuente y referencia bibliográfica utilizada deberá documentarse según lo indica el siguiente apartado, de lo contrario se considerará como fraude. Se recomienda hacer uso de las bases de datos digitales de la biblioteca en el sitio Web <https://www.tec.ac.cr/sistema-bibliotecas-tec>
6. Estándares de codificación para proyectos programados: Es obligación aplicar en el proyecto alguna notación de programación reconocida para la definición de nombres de funciones, variables y otros elementos del lenguaje que se maneje, por ejemplo:
  1. Notación CamelCase:
    - LowerCamelCase: Cada variable deberá definirse con su primer palabra en minúscula y las siguientes con su primer letra en mayúsculas.  
Ejemplos: primerNombre, numeroPuestos, correoPersonal.



- UpperCamelCase: Cada variable deberá definirse con la primer letra de sus palabras en mayúscula.

Ejemplos: PrimerNombre, NumeroPuestos, CorreoPersonal.

2. Notación SnakeCase: Cada una de las palabras estarán unidas con un guion bajo (\_)

- Ejemplos: primer\_nombre, numero\_puestos, correo\_personal

3. Notación Húngara: Es similar a la notación CamelCase, pero agrega un prefijo antes del nombre.

Ejemplo: intContador, chkPuesto, frmPrincipal



# Documentación

La documentación no tienen valor porcentual en el curso de Taller de programación pero si en el curso de Introducción a la Programación. A continuación se describen los apartados básicos que el estudiante deberá presentar en las documentaciones de sus proyectos programados.

## Portada

[Se deberá seguir el estándar siguiente

*Instituto Tecnológico de Costa Rica*

*Unidad de Computación*

***["Título del trabajo"]***

*[Nombre del estudiante]*

*Sede San Carlos*

*[Fecha]*

]

## Citas

*[Todo documento de investigación o desarrollo requiere de fuentes fiables, si utilizamos como referencia trabajos verdaderos y originales; debemos reconocer el mérito y esfuerzo de aquéllos de quienes tomamos prestada la información.*

*El estilo APA requiere que el autor del trabajo documente su estudio a través del texto, identificando el autor y la fecha de los recursos investigados. Este método debe citar por*



*autor-fecha (apellido y fecha de publicación), permite al lector localizar la fuente de información en orden alfabético, en la lista de referencias al final del trabajo.*

*Ejemplos:*

*\*Nota: procesadores de palabras como Word 2007 las hace. En la página del TEC también hay programas para hacer las referencias como es el caso de refworks disponible en la biblioteca del TEC. En el caso de fuentes cuya ficha bibliográfica consigna los dos apellidos se deben anotar los dos apellidos seguidos por un guion.*

*Por ejemplo: Rojas-Pérez, J (2013)*

### **Libro**

*Apellido del autor, inicial del nombre. (Año de publicación) Nombre del libro. (Edición). Ciudad, País: Editorial*

Cassany, D. (2000). *La cocina de la escritura* (Novena ed.). Barcelona, España: Anagrama.

### **Revista**

*Apellido del autor, inicial del nombre. (Año de publicación). Nombre del artículo. Nombre de las revista. Volumen. (Número), páginas en las que aparece el artículo.*

Salazar Bemúdez, G. (2006). Experiencia docente de la Universidad de Costa Rica en el uso de Puntos de Función y metodologías orientadas a objetos para estimar proyectos de software. *Ingeniería* , 16 (2), 116-127.

### **Artículo de periódico**

*Apellido del autor, inicial del nombre. (Día, mes y año de publicación). Nombre del artículo. Nombre del periódico, página.*

Gallardo, H. (9 de Marzo de 2011). Los Arias. *Semanario Universidad* , pág. 20.

### **Sitio web**

*Autor corporativo*

*Real Academia Española. (9 de Marzo de 2011). Real Academia Española. Recuperado el 14 de abril de 2010, de [www.rae.es](http://www.rae.es)*

### **Autor**

Bailey, R. (2011, Marzo 9). The Royal Society. Retrieved Marzo 9, 2011, from The Chemical Basis of Morphogenesis: <http://rstb.royalsocietypublishing.org/content/237/641/37.abstract>

*En la siguiente dirección <http://www.apastyle.org/> encontrara la guía completa del estilo APA.]*





## Referencias

*[Citar todos los documentos que utilizó para elaborar el proyecto. Utilice el formato APA, ordenar alfabéticamente. (Ver ejemplo)*

## Referencias bibliográficas

*Gallardo, H. (2006). Elementos de investigación académica. San José, Costa Rica: EUNED.*

*Hernández Sampieri, R; Fernández, C; Baptista, P (1997) Metodología de la Investigación. México: Mc Graw Hill.*

*Muller, V. (2002) Guía para la elaboración de tesis. San José, Costa Rica: EUCR*

*Tamayo, M. (1999) Serie aprender a investigar: módulo 5 el proyecto de investigación. Colombia: Instituto Colombiano para el fomento de la educación superior.*

*]*

## Introducción

*[Se realiza una breve presentación de la temática a tratar en el proyecto o asignación.*

*“Debe” responder directamente al porqué se realiza el proyecto o asignación. Resulta útil incluir información sobre las últimas tendencias o enfoques aplicados en el área al que pertenece la asignatura.*

*Por otra parte, debe evitarse a toda costa:*

- 1. Redundar nuevamente en los puntos indicados por parte del docente en el proyecto*
- 2. Análisis de tipo filosófico, que introduzcan aspectos que no se relacionan directamente con el proyecto, por ejemplo:*

*“Desde el inicio, el ser humano ha emprendido una búsqueda incansable en pos de convertirse...”*

*“En la actualidad las Tecnologías de la Información juegan un papel importante en cualquier ámbito de la sociedad...”*

*Se pueden presentar ciertos aspectos que evidencian la estructura del informe o artículo, sin entrar necesariamente en aspectos de fondo]*



## Análisis del problema

*[El análisis del problema constituye una herramienta sumamente útil para formar en los alumnos la cultura de la buena comunicación en el ámbito de la Ingeniería de Software.*

*Resulta conveniente analizar los puntos a tratar en la evaluación, considerando estos como requerimientos de un proyecto de software o como las preguntas o situaciones a resolver en un proceso de indagatoria (investigación).*

*No obstante, se deben considerar mecanismos que permitan llevar a cabo una descripción eficiente de la situación y las metodologías o medios necesarios para brindar una posible solución o abordaje al problema.*

*Para ello, se pueden considerar: diagramas de conceptos, técnicas, herramientas, o ejemplos prácticos de situaciones en donde se visualice la problemática o situación planteada, evitando a toda costa la utilización de descripciones o prosa confusa y poco relevante sobre el punto a tratar. Un ejemplo incorrecto sería:*

*“1. Implementar un analizador de texto, para determinar el idioma. Aquí los alumnos deben indagar sobre cómo analizar texto para determinar el idioma del mismo. Jorge Alfaro Velasco es el encargado de analizar este problema, no obstante, Lorena Valerio también ayudará en la solución, ya que como dice el refrán dos cabezas piensan mejor que una...”*  
*]*

## Solución del problema

*[Por otra parte, en contraposición con el análisis del problema, este apartado representa el planteamiento o enfoque para afrontar la temática, situación o problema asignado.*

*Resulta de suma importancia comunicar de manera eficiente la solución adoptada a partir del análisis realizado.*

*Uno de los mejores ejemplos lo constituyen las páginas o sitios WEB enfocados en la solución de problemas o situaciones en el ámbito del software. Por ejemplo: <http://www.codeproject.com/> o [www.stackoverflow.com](http://www.stackoverflow.com) se utilizan en forma*



frecuente para obtener o poner a disposición información relacionada con un problema en particular. Los problemas se abordan a través de los pasos que permiten llevar a cabo su solución, empleando recursos como:

1. *Etapas de la solución*
2. *Diagramas e imágenes de la solución planteada*
3. *Código e incluso pseudocódigo de la posible solución*

*Debe evitarse la descripción de la solución final implementada para el proyecto, en otras palabras, un resumen escueto de las labores realizadas, y que por lo general se agregan una vez realizado el proyecto o poco antes de entregarlo, de manera tal que este apartado se degrada o convierte en una bitácora de las labores realizadas. Un ejemplo incorrecto sería:*

*“Los alumnos se reunieron e implementaron dos métodos, uno principal con los pasos necesarios para llevar a cabo las operaciones aritméticas y otro que se encarga de imprimir los resultados en consola...”*

*“Método ObtenerDatos() Se encarga de obtener los datos suministrados por el usuario y los transfiere a la capa de negocios...”*

*]*

## **Análisis de resultados**

*[En este apartado se debe buscar el mayor acercamiento entre los alumnos y la forma en que deben comunicarse los avances y resultados de los proyectos a nivel profesional, por lo tanto resulta esencial fomentar la visión de este apartado como un análisis del estado de los requerimientos de un proyecto.*

*Debe ser claro y conciso a la hora de expresar estos resultados, sin llevar a cabo un análisis exhaustivo o comentarios que de alguna manera desvíen al lector (cliente, usuario) del objetivo principal: Determinar el estado actual del proyecto.*

*Por ende, se recomienda contar con alguna tabla o formato que permita determinar de forma rápida las labores realizadas y visualizar un panorama general.*



*A continuación, un ejemplo del formato para expresar los resultados:*

<b>Tarea/Requerimiento</b>	<b>Estado</b>	<b>Observaciones</b>
Gestión de usuarios	Completo	
Ordenamiento de lista	Incompleto	El algoritmo no realiza el ordenamiento en algunos casos

**Nota:** Aunque el uso de este tipo de tabla se considera generalizado, no podría decirse lo mismo sobre los datos o indicadores para determinar el estado de las tareas, por ende, se debe recalcar la importancia de que el docente especifique los indicadores a utilizar, por ejemplo:

1. *Porcentuales (suelen utilizarse más en tareas globales, conformadas por otras, de manera que al final reúnen el porcentaje total de avance)*
2. *Estados:*
  1. *Concluido / No concluido*
  2. *Completo / No completado*

*Los datos a incluir en este análisis deben relacionarse estrictamente con el proyecto, no deben girar en torno a percepciones u opiniones personales de los involucrados]*

## **Conclusiones**

*[Deben dirigirse estrictamente a los resultados obtenidos en el proyecto. Por ejemplo:*

1. *Emitir un criterio respecto al rendimiento de un proceso implementado en contraposición con uno alternativo.*
2. *Pertinencia de la aplicación de un enfoque de programación en algún ámbito en particular de acuerdo a la experiencia o resultados del proyecto.*
3. *Resultados concretos del proyecto, por ejemplo: “Se logró determinar la eficiencia del proceso X con respecto al Y, en los ámbitos que implican...”*

*No deben incluirse aspectos o criterios personales, por ejemplo:*

*“Al concluir este proyecto aprendimos sobre el funcionamiento de las estructuras de datos como: árboles y grafos, y la importancia de estos...”*

*“El actual proyecto nos enseña sobre la importancia de los sistemas de gestión de bases*





*de datos en el ámbito empresarial...”*

*]*

## **Recomendaciones**

*[Al igual que las conclusiones, se trata de un espacio en donde se pueden plantear:*

- 1. Aspectos o ámbitos de interés para ampliar el alcance del proyecto.*
- 2. Áreas o campos de estudio que pueden complementar las temáticas analizadas.*
- 3. Ideas que sean de utilidad para interesados en el área en  
que se desenvuelve el proyecto.*

*No se deben abordar temáticas o aspectos relacionados con la percepción de los integrantes, y que en general no aportan en absoluto al tema principal, por ejemplo:*

*“Se considera que el tiempo para realizar el proyecto fue...”*

*“Se recomienda que el profesor brinde una explicación sobre temas...”*

*]*

