

GUÍA PARA LA REALIZACIÓN DEL PROYECTO FINAL DE ESTRUCTURAS DE DATOS 2

Texto Para Estudiantes



Sección 1: Requerimientos para la realización de las entregas del proyecto final	<u>Pág.3</u>
Sección 2: Intercambio de archivos	<u>Pág.5</u>
Sección 3: Porcentajes y tiempos de entrega del proyecto	<u>Pág. 7</u>
Sección 4: Primera Entrega	<u>Pág. 9</u>
Sección 5: Segunda Entrega	<u>Pág. 12</u>
Sección 6: Tercera Entrega	<u>Pág. 28</u>
Sección 7: Sustentación Oral	<u>Pág. 35</u>
Sección 8: Prácticas para el desarrollo	<u>Pág. 37</u>
Sección 9: Rúbricas de Calificación	<u>Pág. 38</u>

Requerimientos para la realización del proyecto final

El proyecto final del curso será desarrollado en 3 entregas que se explicarán en detalle en la presente guía. **A continuación, lea atentamente las consideraciones a seguir para el envío de cada una de ellas:**



El proyecto será **desarrollado en parejas y cada uno debe dar cuenta de todo** lo que entregan en códigos e informes



Preferiblemente **escribirlo en inglés**

NOTA: Eviten el uso de abreviaciones, tales como, *there're* y *aren't*. En su lugar, **escriban las palabras completas**, *there are* y *are not*, para el ejemplo



Usen la **plantilla** de la **Association for Computing Machinery (ACM)** que entrega el docente

NOTA: No usen normas APA o ICONTEC



No usen alineación a la izquierda, derecha o al centro, en los textos. **Todos deben de ir justificados**



Referenciar las fuentes bibliográficas y cibergráficas usando el formato para referencias de la **ACM**. Léase en <http://bit.ly/2pZnE5g>



El informe debe ser entregado en PDF y el código en GIT



El informe en PDF deben escribirlo en **máximo 4 páginas** de extensión



Para efectos de **referenciación de códigos** véase la “*Guía Metodológica para la Realización y Entrega de los Laboratorios de Estructuras de Datos y Algoritmos*” en la Sección 4, numerales 4.16 y 4.17

Para no incurrir en el reglamento del curso, **Véase el Código de Ética** que encuentra en la Sección 1 de la misma

Intercambio de archivos entre el docente y los estudiantes:



A los estudiantes, el docente les entregará un archivo ZIP, que contiene:

- ☒ Un documento PDF con las especificaciones del proyecto,
- ☒ Un documento PDF con la presente guía para la realización del Proyecto Final de Estructura de Datos 2
- ☒ Plantilla ACM para la realización del informe (Word y Latex),
- ☒ Plantilla en PowerPoint para hacer las diapositivas,
- ☒ Ejemplo en español y en inglés de un informe en plantilla ACM,
- ☒ Ejemplo en español y en inglés de las diapositivas

Por su parte, **los estudiantes deberán hacer tres entregas al docente**, cada una en un archivo ZIP que contiene:

- ☒ **Primera entrega:** Un documento PDF con el informe usando plantilla ACM
- ☒ **Segunda entrega:** Un documento PDF con el informe usando plantilla ACM y un código debidamente documentado en GIT

- ☑ **Tercera entrega:** Un documento PDF con el informe usando plantilla ACM (opcionalmente con Anexos de trabajo en equipo), un documento PDF con las diapositivas usando la plantilla institucional (opcionalmente con el link al reporte en *arxiv*) y un código debidamente documentado en GIT

Entiendan lo anterior así:



Porcentajes, criterios de evaluación y tiempos de entrega del proyecto

Cada entrega del Proyecto Final del curso, suma un porcentaje de valoración específico, cuenta con un tiempo de trabajo determinado y considera requerimientos particulares, a seguir

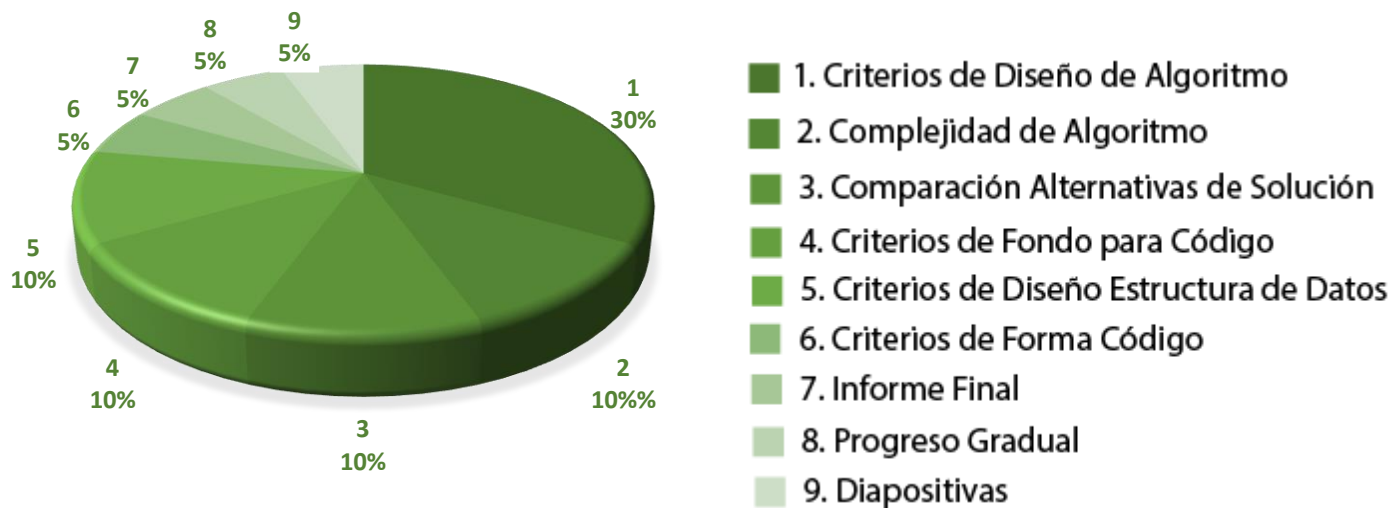


En la vida real, si un desarrollo de software no se termina, no se paga. Es decir, si se hacen entregas parciales, se hace la documentación, se trabaja en equipo; pero no se entrega una solución, no se paga el desarrollo.

No obstante, si no se cumple con las entregas parciales, no se hace una buena documentación o no se entrega a tiempo, pero se termina el desarrollo, se paga, pero se cobran multas.

Teniendo en cuenta lo anterior, para la evaluación del proyecto, si hacen las primeras dos entregas, pero no la última, la nota es de 0.0; pero si hacen la última entrega, sin hacer las dos primeras, se califica la práctica final sobre 4.5

Porcentajes y criterios de evaluación:



Tiempos de entrega en semanas académicas:

El semestre académico tiene una extensión de 16 semanas, y, por tanto, es necesario distribuir las entregas del proyecto final en el transcurso del mismo, así:



Primera Entrega

DOCUMENTACIÓN DE 4 PROBLEMAS SIMILARES (0%):

En el informe PDF:

a) **Título:** Descripción del proyecto. Entre 8 y 12 palabras.



PISTA: La siguiente guía muestra cómo escribir un buen título
<http://bit.ly/2mFqHeC>

b) **Autores:** Escriban sus nombres y los datos académicos que solicita la plantilla. Incluyan de último, el nombre del docente

c) **Introducción:** Es la justificación de las condiciones en el mundo real que llevan al problema

d) **Problema:** En pocas palabras escriban cuál es el problema que hay que resolver

e) **Problemas similares y sus soluciones:** Expliquen 4 problemas algorítmicos similares que se encuentren documentados en libros, artículos científicos o sitios web, y al menos 1 solución para uno de ellos



PISTA: Eviten incurrir en los siguientes errores comunes: (I) explicar el problema y no la solución. (II) Explicar la solución y no el problema. (III) Problemas de tecnologías similares (Por ejemplo, JQuery)



PISTA 2: Observen a continuación un ejemplo correcto y otro incorrecto de lo que se pide hacer en esta entrega

Ejemplo Correcto	
Tema	Problema Similar
3.2 Double metaphone	
Algoritmo para un sistema de texto predictivo e indexación de nombres por sonido	<p>“Double metaphone” es un algoritmo fonético publicado por Lawrence Philips en 1990 para indexar palabras y nombres con base en su pronunciación inglesa [3], utilizando dos conjuntos de reglas para codificar las palabras fonéticamente en cada uno de ellos, uno para el inglés y otro para una representación alternativa. Se diferencia de su predecesor “Metaphone” debido a que retorna ambos valores (de ahí la palabra “Double”) [4], eliminando así algunas ambigüedades con las que “Metaphone” no podía lidiar.</p>
Ejemplo Incorrecto	
Tema	Problema Similar
3.4 Autocompletado en JQuery	
Algoritmo para un sistema de texto predictivo e indexación de nombres por sonido	<p>En el algoritmo de autocompletado de JQuery, implementado principalmente en páginas web, los datos sobre los que se busca (para cantidades grandes de datos) se ubican en un URL externo, de modo que cuando se ingresa un input, una solicitud GET es enviada a ese url; a manera de ejemplo, si el url especificado es www.ejemplo.com, al ser tecleada la palabra “hol”, la solicitud será <code>ejemplo.com?term=hol</code>, se creará una <i>expresión regular</i> (herramienta para encontrar texto de acuerdo a un patrón[11]) y se mostrarán las palabras que, dentro de la base de datos, posean el mismo patrón[9].</p>



f) Gráficas de los problemas similares y sus soluciones: Usen gráficas vectorizadas para explicar los problemas similares y sus soluciones



PISTA: Para saber qué son los gráficos vectorizados y sus ventajas lean en <http://bit.ly/2pBn6lO>



PISTA 2: Se recomienda el uso de *Inkscape* o *Adobe Illustrator* para vectorizar los gráficos

g) Referencias de los problemas similares y sus soluciones: Referencien los aportes bibliográficos que utilizaron en el numeral e), según el formato ACM

h) [Anexo Opcional]: Incluir la documentación de progreso gradual y trabajo en equipo en el Anexo del Informe PDF:

- ☒ Actas
- ☒ Control de versiones del código
- ☒ Control de versiones del documento del informe



NOTA: La extensión de este anexo no suma en las 4 páginas máximas permitidas del Informe PDF



PISTA: Véase “*Guía Metodológica para la Realización y Entrega de los Laboratorios de Estructura de Datos y Algoritmos*” en la Sección 4, numeral 4.21

Segunda Entrega

IMPLEMENTACIÓN DE LA PRIMERA VERSIÓN DE LA SOLUCIÓN DEL PROBLEMA (0%):

El objetivo de esta es entregar la primera versión de la solución del problema que ustedes proponen a partir de la definición de sus propios algoritmos.

Las actividades a realizar para esta entrega son:

En el informe PDF:

a) Palabras claves del autor (*Author Keywords*): Estas son palabras claves que ustedes consideran apropiadas para indexar el informe PDF en bibliotecas o bases de datos



En la vida real, el esquema de palabras claves de la ACM, permite clasificar artículos de revistas y conferencias de la ACM en su biblioteca digital para permitir búsquedas más descriptivas. Más detalles en <http://www.acm.org/about-acm/class>

b) Palabras claves de la ACM: Solo las que están en esta lista: <http://www.acm.org/about/class/2012> No pueden ser inventadas.



PISTA: Vean a continuación ejemplos de palabras claves usando el esquema jerárquico de la ACM

Ejemplo 1: *Theory of computation → Design and analysis of algorithms → Graph algorithms analysis → Shortest paths*

Ejemplo 2: *Theory of computation → Design and analysis of algorithms → Data structures design and analysis → Online algorithms*

c) Todo lo que hicieron en la Entrega 1: Introducción, Problema, Problemas Similares, Gráficas y Referencias

d) Diseño de la estructura de datos: Diseñen la estructura de datos para resolver el problema y gráfíquenlo. No usen gráficas extraídas de internet

Sección 5: Segunda Entrega



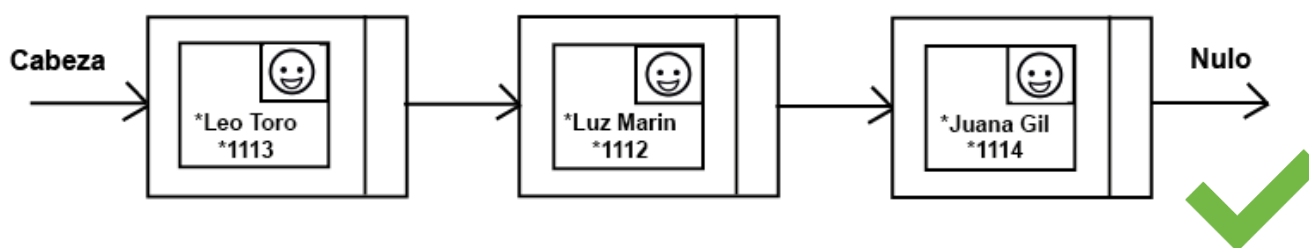
NOTA: En este numeral incluyan el enlace al repositorio de la nube en donde se encuentra el código del proyecto



PISTA: Observen a continuación un ejemplo correcto y otro incorrecto de lo que se pide hacer en este numeral

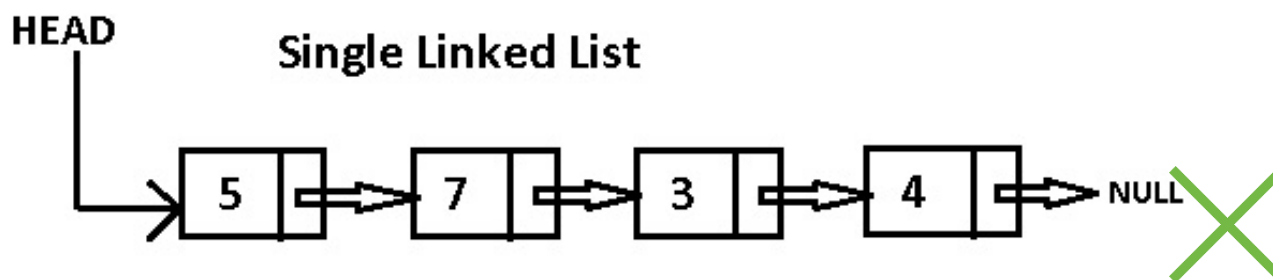
Ejemplo Correcto

Diseñar una estructura de datos para almacenar eficientemente personas



Ejemplo Incorrecto

Lista simplemente encadenada extraída de internet





PISTA: Pueden apoyarse en libros, artículos científicos, sitios web y relatos anecdóticos de expertos

e) Diseño de las operaciones de la estructura de datos: Diseñen las operaciones de la estructura de datos para solucionar el problema eficientemente.

Usen la estructura de datos que diseñaron e incluyan una imagen explicando cada operación. **Las imágenes deben ser vectorizadas**



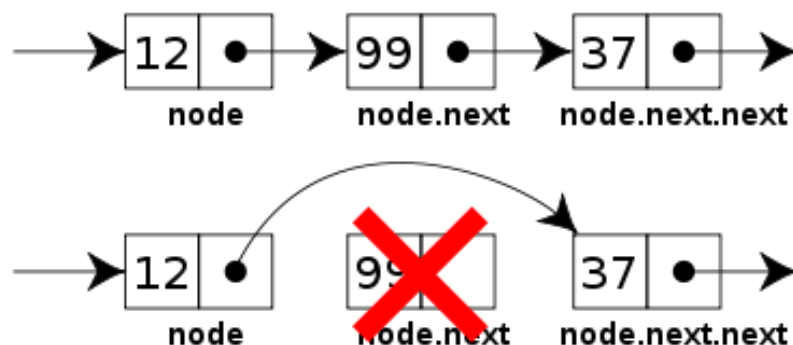
NOTA: Deben explicar los criterios de diseño que tuvieron en cuenta para diseñar la estructura de datos principal



PISTA: Observen a continuación un ejemplo correcto y otro incorrecto de lo que se pide en este numeral

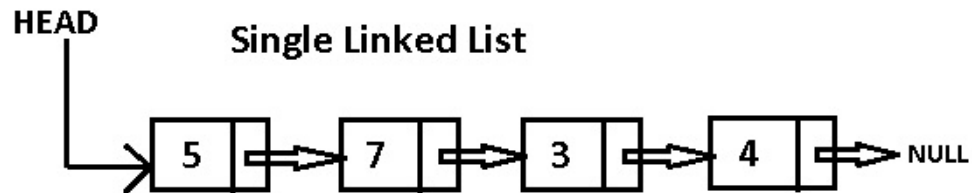
Ejemplo Correcto

Imagen de una operación de borrado de una lista encadenada



Ejemplo Incorrecto

Lista simplemente encadenada extraída de internet



f) Cálculo de las operaciones de la estructura de datos: Calculen la complejidad de las operaciones de la estructura de datos para el peor de los casos, el mejor de los casos y el caso promedio



PISTA: Observen a continuación un ejemplo para reportar la complejidad

Tabla de Complejidades

Método	Complejidad
Búsqueda Fonética	$O(1)$
Imprimir búsqueda fonética	$O(m)$
Insertar palabra búsqueda fonética	$O(1)$
Búsqueda autocompletado	$O(s + t)$
Insertar palabra en TrieHash	$O(s)$
Añadir búsqueda	$O(s)$



PISTA 2: Apóyense en el sitio <http://bigocheatsheet.com/> para ampliar información sobre el cálculo de complejidad

g) Criterios de diseño de la estructura de datos: Expliquen por qué diseñaron así la estructura de datos



NOTA: Recuerden que este es el numeral con mayor porcentaje de evaluación



PISTA: Los criterios **deben ser objetivos**, por ejemplo, la eficiencia en tiempo y memoria, o algunos como los que exponen a continuación:

A la hora de decidir qué estructura de datos usamos en la comparación fonética hay que tener en cuenta que estamos haciendo una relación entre el Código fonético de la palabra y la palabra misma, por lo cual la estructura de datos debe tener en cuenta ese mapeo. En ese orden de ideas, usamos una HashTable (o HashMap en java), las **Keys** de la tabla serán todos los códigos fonéticos de las palabras del diccionario, y cómo **Value** tendremos todas las palabras a las cuales coinciden con su respectivo código fonético, que serán almacenadas en una LinkedList (Se usó una LinkedList, porque permite el uso de iteradores lo cual aumentaba su rendimiento a la hora de recorrerla para su impresión). Esta estructura nos permite obtener, a partir del

No deben usar criterios no objetivos, tales como: “me enfermé”, “fue la primera que encontré”, “la hice el último día”, “ese tema no lo entendí”, “no vine a clase”, **estos rebajarán la nota**

h) Diseño del algoritmo: Diseñen el algoritmo para resolver el problema y gráfiquenlo. No usen gráficas extraídas de internet

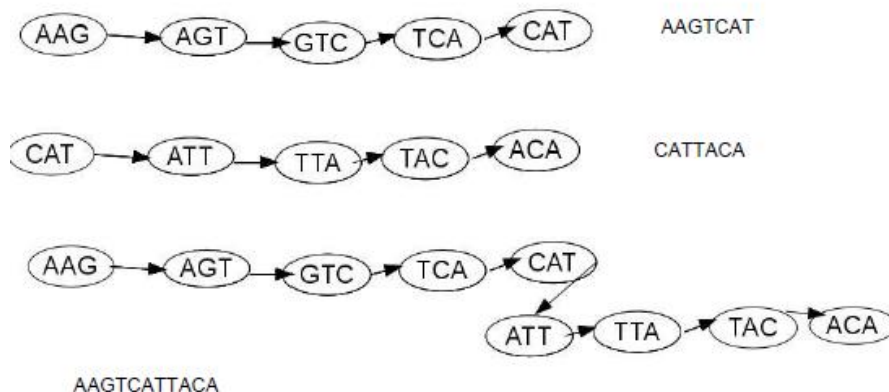


PISTA: Observen a continuación un ejemplo correcto y otro incorrecto de lo que se pide hacer en este numeral

Ejemplo Correcto

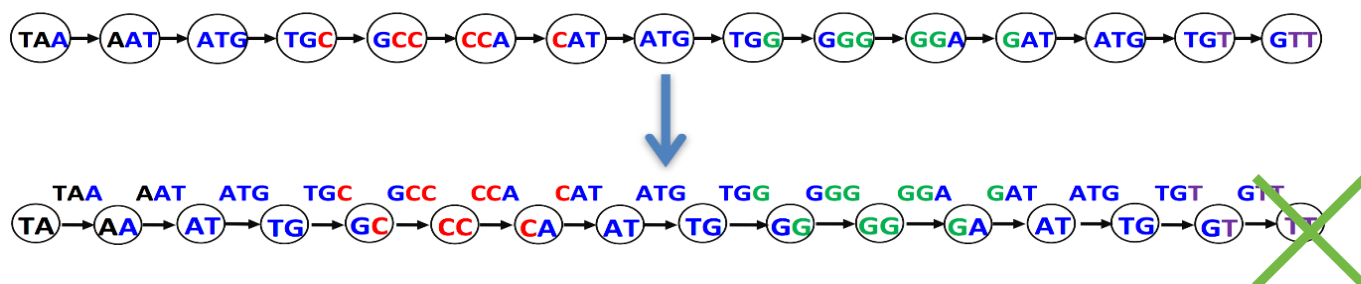
A continuación, se explica paso a paso cómo se ensamblan fragmentos de ADN que entrega un secuenciador de ADN en la cadena original, utilizando como estructura de datos los grafos de Bruijn

Para el caso de la cadenas:



Ejemplo Incorrecto

Ejemplo extraído de internet donde no es claro cómo se aplica eso al problema del proyecto



PISTA: Pueden apoyarse en libros, artículos científicos, sitios web y relatos anecdóticos de expertos


i) Cálculo de la complejidad del algoritmo: Calculen la complejidad del algoritmo para el peor de los casos, el mejor de los casos y el caso promedio.



PISTA 2: Observen a continuación un ejemplo correcto e incorrecto de lo que se pide hacer para este numeral

Ejemplo Correcto

A continuación, se describe la complejidad de cada uno de los sub problemas que componen el algoritmo, finalmente, calculamos la complejidad total. Sea A la longitud de una secuencia de ADN, N el número de secuencias de ADN, y V el número de K-meros diferentes que se obtienen de las secuencias de ADN.

Sub problema	Complejidad	
Crear el grafo de Bruijn con las secuencias	$O(N)$	
Actualizar el grafo de Bruijn con las secuencias	$O(A \cdot N^2)$	
Encontrar los genes	$O(V)$	
Complejidad Total	$O(A \cdot N^2 + V)$	

Ejemplo Incorrecto

Un ejemplo incorrecto se determinó la complejidad de cada subproblema que compone el algoritmo, pero no se calculó la complejidad total. Además, no es claro quién es la variable “n”.

Sub problema	Complejidad	
kmers	$O(n)$	
tieneInicio	$O(n^2)$	
visitar	$O(n)$	
generarGen	$O(n)$	

j) Criterios de diseño del algoritmo: Expliquen por qué diseñaron así el algoritmo



NOTA: Recuerden que este es el numeral con mayor porcentaje de evaluación



PISTA: Los criterios **deben ser objetivos**, por ejemplo, la eficiencia en tiempo y memoria, o algunos como los que exponen a continuación:

After analyzing the different solutions for the problem, we conclude that the de Bruijn Graph is an optimal option to reconstruct the sequence due to its ease to traverse and its integrity about maintaining the original DNA string, and Boyer-Moore offers a quickly way to find codons inside the DNA, in a way that area easily processed to be defined as a gen or no, this on-line solution is viable because sequence with mutation or reading errors were not considered.

No deben usar criterios no objetivos, tales como: “me enfermé”, “fue la primera que encontré”, “la hice el último día”, “ese tema no lo entendí”, “no vine a clase”, **estos rebajarán la nota**

k) Resultados obtenidos de la primera solución: Calculen, (I) el tiempo de ejecución y (II) la memoria usada del algoritmo, para el *Conjunto de Datos* que está en el ZIP:

- ☑ Para cada conjunto de datos, tomen el tiempo 100 veces. Esto no lo reporten
- ☑ Para cada conjunto de datos, coloquen en el tiempo promedio de esas 100 veces. Esto sí lo reportan.
- ☑ Para los tiempos promedio de cada conjunto de datos, saquen el más largo, el más corto y el promedio. Esto sí lo reportan.



PISTA: Observen a continuación un ejemplo para calcular el tiempo 100 veces y la memoria utilizada

Paso 0: Cierren todos los programas y no ejecuten otros mientras se toman los tiempos.

Paso 1: Tomen 100 veces el tiempo de ejecución y memoria de ejecución, para cada conjunto de datos y para cada operación del algoritmo.

Reporten en la tabla la siguiente información:

Tabla 1. *Tiempos de ejecución del algoritmo con diferentes conjuntos de datos*

	Conjunto de Datos 1	Conjunto de Datos 2	...Conjunto de Datos n
<i>Mejor caso</i>	10 sg	20 sg	5 sg
<i>Caso promedio</i>	12 sg	10 sg	35 sg
<i>Peor caso</i>	15 sg	21 sg	35 sg

Tabla 2. Consumo de memoria del algoritmo con diferentes conjuntos de datos

	Conjunto de Datos 1	Conjunto de Datos 2	...Conjunto de Datos n
Consumo de memoria	10 MB	20 MB	5 MB

Paso 2: Calculen el tiempo y memoria, mínimo, máximo y promedio del algoritmo

	Mejor tiempo	Peor tiempo	Tiempo promedio	Mejor memoria	Peor memoria	Memoria promedio
Mejor caso	5 sg	20 sg	15 sg	5 MB	20 MB	14 MB
Caso promedio	5	20	...			
Peor caso						

Nota: Estas tablas no tienen valor sin las unidades de medida. Por ejemplo, los tiempos se miden en segundos o milisegundos y la memoria en megabytes, gigabytes o bytes.

Paso 3: Expliquen los resultados

Como un ejemplo, el algoritmo *Quicksort* puede tomar en el peor de los casos 1 minuto para ordenar un arreglo de 100'000.000; pero en el mejor de los casos toma 10 segundos y en el caso promedio 35 segundos, entonces, es un buen algoritmo para el caso promedio, aunque haya unos casos en los que tome mucho tiempo.

De la teoría, sabemos que la complejidad en el peor de los casos es $O(n^2)$ mientras que en el mejor de los casos y el caso promedio es $O(n \cdot \log n)$.

Otro ejemplo, la estructura de datos que diseñamos es muy rápida para buscar y para insertar, y lenta para borrar, pero para el problema que estamos resolviendo no se necesita borrar casi nunca, entonces no hay inconveniente.



PISTA 2: Observen a continuación ejemplos correctos de las tablas que se piden para este numeral

Ejemplo 1

La siguiente tabla muestra los tiempos de ejecución de un algoritmo de reconstrucción de ADN con el ADN mitocondrial 3 especies de vertebrados que se encontraban en el conjunto de datos.

	Mejor (ms)	Tiempo Peor (ms)	Tiempo Promedio (ms)
Acanthisitta chloris	134188	156654	144192
Acipenser transmontanus	115673	130525	119688
Anoplogaster cornuta	66031	93784	73517

Ejemplo 2

La siguiente tabla muestra el consumo de memoria de un algoritmo de reconstrucción de ADN con el ADN mitocondrial 3 especies de vertebrados que se encontraban en el conjunto de datos.

	Mejor memoria (bytes)	Peor memoria (bytes)	Memoria promedio (bytes)
Acanthisitta chloris	10137400	131168816	270298
Acipenser transmontanus	9452437	120912192	130364
Anoplogaster cornuta	8521258	115983521	124504

I) **[Anexo Opcional]** Completen el Trabajo en Equipo y Proceso Gradual que adelantaron desde el principio del semestre



PISTA: Véase “*Guía Metodológica para la Realización y Entrega de los Laboratorios de Estructura de Datos y Algoritmos*” en la Sección 4, numeral 4.21

En el código entregado en GIT:

m) Solución final al problema: Implementen un algoritmo para solucionar el problema.

Además, pruébenlo con los datos que están en la carpeta de *Conjunto de Datos* del .ZIP

n) Documentación en JavaDoc: Incluyan la documentación en JavaDoc



PISTA: Véase “*Guía Metodológica para la Realización y Entrega de los Laboratorios de Estructura de Datos y Algoritmos*” en la Sección 4, numeral 4.1

En otro documento en formato PDF:

o) Presentación en diapositivas: Entreguen las diapositivas para presentar el proyecto usando la plantilla *Eafit* que les entrega el docente (*Máximo 6 diapositivas*)

A continuación, el contenido a incluir dentro de cada diapositiva:

☒ **Diapositiva 1:** Título del proyecto e integrantes

☒ **Diapositiva 2:** Estructura de datos diseñada, incluyendo las imágenes que la explica.

☒ **Diapositiva 3:** Explicación del algoritmo y su complejidad.

- ☑ **Diapositiva 4:** Por qué el algoritmo diseñado es una bueno para la solución. Incluyan sólo argumentos objetivos.
- ☑ **Diapositiva 5:** Incluyan las gráficas de los tiempos y memoria obtenidos con el algoritmo.
- ☑ **Diapositiva 6:** Imagen del software en funcionamiento
- ☑ **[Opcional]: Diapositiva 7** Incluir el enlace o url en *arXiv* donde quedó el reporte aceptado.



PISTA: Vean la presentación ejemplo que se les entrega junto al proyecto

Tercera Entrega o Informe Final

IMPLEMENTACIÓN DE LA VERSIÓN FINAL DE LA SOLUCIÓN DEL PROBLEMA (100%):

El objetivo es mostrar la solución al problema y reportarla según los requerimientos técnicos de la ACM

Las actividades a realizar para esta entrega son:

En el informe PDF:

a) Resumen: Para escribirlo pueden dar respuesta a estas preguntas: ¿Cuál es el problema?, ¿Por qué es importante el problema?, ¿Qué problemas relacionados hay?, ¿Cuál es la solución?, ¿Cuáles son los resultados? y ¿Cuáles las conclusiones? Utilizar máximo 200 palabras.

b) Todo lo que hicieron en la Entrega 1 y Entrega 2: Introducción, Problema, Problemas Similares, Gráficas, Referencias, Palabras claves del autor, Palabras claves de la ACM, Estructura de Datos Principal, Resultados obtenidos

c) Diseño de la estructura de datos: Diseñen la nueva estructura de datos para resolver el problema y gráfiquenlo, tal y como lo hicieron en el numeral *d)* de la *Segunda Entrega*. No usen gráficas extraídas de internet

d) Diseño de las operaciones de la estructura de datos: Diseñen las operaciones de la nueva estructura de datos para solucionar el problema eficientemente, tal y como lo hicieron en el numeral *e)* de la *Segunda Entrega*.

e) Cálculo de complejidades de las operaciones de las estructuras de datos:

Calculen la complejidad del algoritmo, tal y como lo hicieron en el numeral f) de la *Segunda Entrega*

f) Criterios de diseño de la estructura de datos: Expliquen por qué diseñaron así la nueva estructura de datos, tal y como lo hicieron en el numeral g) de la *Segunda Entrega*

g) Diseño del algoritmo: Expliquen el nuevo algoritmo, tal y como lo hicieron en el numeral h) de la *Segunda Entrega*



PISTA: Incluyan imágenes que expliquen la estructura de datos

h) Cálculo de la complejidad del algoritmo: Calculen la complejidad del algoritmo para el peor de los casos, el mejor de los casos y el caso promedio, tal y como lo hicieron en el numeral i) de la *Segunda Entrega*

i) Criterios de diseño del algoritmo: Expliquen por qué diseñaron así el nuevo algoritmo



NOTA: Tengan en cuenta las instrucciones dadas del numeral j) de la *Segunda Entrega*

j) Resultados obtenidos de la solución final: Calculen, (I) el tiempo de ejecución y (II) la memoria usada para las operaciones del nuevo algoritmo, para el *Conjunto de Datos* que está **en el ZIP**



NOTA: Consideren las pistas dadas sobre esta actividad en el numeral k) de la Segunda Entrega

k) Conclusiones: Léase a continuación un texto sobre cómo deben escribirlas

Para escribir las conclusiones, se procede de la siguiente forma;

- ☒ En un párrafo escriban un resumen de lo más importante que hablaron en el reporte
- ☒ En otro expliquen los resultados más importantes, por ejemplo, los que se obtuvieron con la solución final.
- ☒ Luego, comparen la primera solución que hicieron con los trabajos relacionados y la solución final.
- ☒ Por último, expliquen los trabajos futuros para una posible continuación de este proyecto

Nota: Cada uno de estos apartes deben escribirlo en un párrafo de máximo 6 líneas

I) Agradecimientos: Lean a continuación un texto sobre cómo deben escribirlo

Para escribir los agradecimientos, se procede de la siguiente forma:

I. Identifiquen el tipo de agradecimiento que van a escribir. Existen 2 tipos de agradecimientos:

- ☒ Agradecimiento a una institución que paga por sus estudios (por ejemplo, Fondo EPM)
- ☒ Agradecimiento a una persona (fuera del Docente porque es autor) que le ayudó con el manuscrito o el código

II. Escriben el agradecimiento según el idioma

☒ **Agradecimiento en inglés:**

This research was supported/partially supported by [Name of Foundation, Grant maker, Donor].

We thank for assistance with [particular technique, methodology] to [Name Surname, position, institution name] for comments that greatly improved the manuscript.

☒ **Agradecimiento en español:**

Esta investigación fue soportada parcialmente por [Nombre de la fundación].

Nosotros agradecemos por la ayuda con [una técnica, metodología] a [Nombre, apellido, posición (digamos Docente), nombre de la institución] por los comentarios que nos hizo para mejorar el [informe o código]

Tengan en cuenta que:

- ☒ El nombre del docente no va porque él es autor. Tampoco sitios de internet ni autores de artículo leídos con quienes no se han contactado
- ☒ Los nombres que sí van son quienes ayudaron, compañeros del curso o docentes de otros cursos.

m) Referencias: Completar las referencias y ajustarlas, si no lo han hecho, al formato ACM.



PISTA: Para referenciar las fuentes usando el formato para referencias de la **ACM** léase en <http://bit.ly/2pZnE5g>

n) [Anexo Opcional] Trabajo en equipo: Completen el Trabajo en Equipo y Proceso Gradual que adelantaron en Entrega 1 y Entrega 2, es decir, muestren lo qué hicieron desde que inició el semestre académico, pasando por lo adelantado en dichas entregas.

En el código comprimido en ZIP:

o) Solución final del problema Implementen un algoritmo para solucionar el problema. Además, pruébenla con los datos que están en la carpeta de *Conjunto de Datos* del .ZIP

p) Documentación en JavaDoc: Incluyan la documentación en JavaDoc



PISTA: Véase “*Guía Metodológica para la Realización y Entrega de los Laboratorios de Estructura de Datos y Algoritmos*” en la Sección 4, numeral 4.1

En otro documento en formato PDF:

q) Presentación en diapositivas: Completen las diapositivas teniendo en cuenta los resultados obtenidos en la entrega final



NOTA: Para recordar el contenido que deben incluir en las diapositivas vean el numeral 1) de la Segunda Entrega.

r) [Ejercicio Opcional] Reporte en arXiv: El reporte es aceptado en el repositorio mundial de reportes de ingeniería de sistemas, arXiv (<http://arxiv.org/>).

En una diapositiva adicional de la presentación mencionada en el numeral inmediatamente anterior, incluir el enlace o url en *arXiv* donde quedó el reporte aceptado.

Sustentación Oral



En la vida real, un ingeniero debe saber presentar su trabajo en forma sintética. Expresarse oralmente o por escrito en su idioma y/o en inglés, es indispensable. El ingeniero de sistemas debe adaptarse a las circunstancias de tiempo y público.

Lo anterior significa ser capaz de presentar en dos minutos, 20 o dos horas, en 10 líneas o en 100 páginas, adaptando el discurso a un público de expertos, clientes o neófitos

Tomado de <http://bit.ly/2p92vUv>

A continuación, lean las indicaciones a seguir para la sustentación oral del proyecto final que se realiza en el aula de clase y la sala de computadores habitual del curso, durante la semana 16:

Tiempos y actividades sugeridos:

Tiempo en Minutos	Actividades sugeridas	Diapositiva
1	Título del proyecto e integrantes	1
1	Explicar el diseño de su estructura de datos (EDA)	2
2	Explicación del algoritmo y su complejidad	3

2	Por qué el algoritmo diseñado es bueno para la solución	4
2	Mostrar su algoritmo funcionando	5
2	Mostrar sus resultados de tiempo y memoria	6
Total 10 Minutos	Responder a preguntas de profesores o monitores	

Reglas importantes:

1. Traer su propio computador
2. El computador traen debe tener una salida a VGA o un adaptador. Si no tienen el adaptador, deben pedir al profesor que les preste uno, 72 horas antes de sustentar.
3. Deben indicar si tienen un *Mac*, *PC* y qué puerto. Ejemplos: *Minidisplay*, *HDMI* y *USB-C*.
4. Preparar la exposición para que dure 9 minutos
5. Legar a tiempo. Si su hora de exposición es entre 9:00 AM a 10:30 AM, llegar a las 9:00 AM al salón. Si su hora de exposición es entre 10:30 AM y 11:45 AM, deben llegar a las 10:30 AM a la sala de computadores

Prácticas para el Desarrollo del Proyecto



Una práctica ágil de desarrollo de *Software*, incentivada por la comunidad *Xtreme Programming*, es la programación en parejas.

Esta práctica consiste en que dos programadores trabajen al mismo tiempo en un solo computador. Uno de los desarrolladores, el conductor, escribe el código; mientras que el otro, el revisor, analiza cada línea de código que se digita y señala posibles errores u optimizaciones.

Después de cierto tiempo, los desarrolladores intercambian roles.

En este proyecto, ustedes deben utilizar esta práctica de desarrollo ágil. El conductor es responsable de subir a la nube, con su cuenta de *git* o *svn*, las actualizaciones que se hicieron mientras él era conductor

Rúbricas de Calificación

Lean sobre el proceso de evaluación, incluyendo criterios de calificación, niveles sobre cada criterio y consideraciones especiales.

%	Criterio	Excelente	Bueno	Malo
10%	Comparación de su solución con otras alternativas de solución	Argumentos objetivos	Argumentos objetivos y subjetivos	Solo argumentos subjetivos o no se hizo
20%	Complejidad del algoritmo	Se calculó correctamente y es apropiado para el problema	Se calculó correctamente pero no es apropiado para el problema	No se calculó
5%	Criterios de Forma para el Código	Doc HTML, acoplamiento, cohesión, indentación, nombre de variables	3 de 5 elementos	Menos de 3
10%	Criterios de Fondo para el Código	La implementación corresponde al diseño de las estructuras de datos	N/A	No corresponde lo implementado a lo diseñado
10%	Criterios de Diseño de la Estructura de Datos	Todos los argumentos son objetivos	Hay argumentos objetivos y otros subjetivos	Todo es subjetivo, no saben o no responden

30%	Criterios de Diseño del Algoritmo	Todos los argumentos son objetivos	Hay argumentos objetivos y otros subjetivos	Todo es subjetivo, no sabe o no responden
5%	Informe Final	Entregar informe según indicaciones dadas		
5%	Progreso Gradual	Hicieron la entrega 1 y 2	Hicieron al menos 1	Ninguna
5%	Diapositivas	Hacer las diapositivas según lo descrito en la Sustentación Oral		
5% Extra	Trabajo en Equipo	Actas de reunión t reporte en <i>git</i> o <i>svn</i> con los cambios que hizo cada uno	Uno de los 2 elementos	Ninguno
10% Extra	Reportes en ArXiv	Aceptado		No Aceptado
5% Extra	Presentación oral e informe en inglés	Sustentación y reporte en inglés	1 de los 2	Ninguno de los dos

Consideraciones especiales para la Evaluación:

- ☒ **Comparación entre alternativas de solución:** Se validarán los argumentos orales que los estudiantes den en el momento de la presentación del trabajo al docente

- ☑ **Complejidad del algoritmo:** Se evaluará que la complejidad esté bien calculada y que sea una complejidad apropiada para el problema que intentan resolver
- ☑ **Criterios de forma para el Código:** Se evaluará que el código esté correctamente indentado, el nombre de las variables sea claro, que tenga comentarios descriptivos y exista una documentación en formato HTML, y que tenga en cuenta el acoplamiento y la cohesión.
- ☑ **Criterios de fondo para el Código:** Se evaluará que la implementación corresponde al diseño de las estructuras de datos y de las operaciones de la estructura de datos
- ☑ **Criterios de Diseño del algoritmo:** Se validará que expliquen cómo lo diseñaron.
- ☑ **Informe final:** Se validará que entreguen el informe en formato PDF.
- ☑ **Progreso gradual:** Tendrá un descuento del 5% de la nota de la práctica, quienes hayan entregado tarde la primera o la segunda entrega, y un descuento del 10% quienes no incluyan en la implementación final y en el informe final, lo solicitado en la primera y en la segunda entrega. Se revisarán las entregas parciales en Interactiva.
- ☑ **Presentación de Diapositivas:** Se validará que entreguen las diapositivas en formato PDF con el formato propuesto. La exposición deberá hacerse en 10 minutos.
- ☑ **[Ejercicio Opcional] Trabajo en equipo:** Se validará que en el Anexo PDF, entreguen copia de todas las actas de reunión y el reporte de git o svn con los cambios en el código y quién los hizo.
- ☑ **[Ejercicio Opcional] Reporte en arXiv:** Se validará que incluyan en el reporte en formato PDF y en las diapositivas, un vínculo al URL donde quedó el reporte.
- ☑ **[Ejercicio Opcional] Reporte y presentación oral en inglés**

¿Alguna inquietud?

CONTACTO

Docente **Mauricio Toro Bermúdez** (W)

Teléfono: (+57) (4) 261 95 00 **Ext.** 9473

Correo: mtorobe@eafit.edu.co

Oficina: 19- 627

Agende una cita con él a través de <http://bit.ly/2gzVg10> , en la pestaña **Semana**. *Si no da clic en esta pestaña, parecerá que toda la agenda estará ocupada.*



Realizadores

Este texto fue escrito y corregido por
Mauricio Toro Bermúdez
Luisa Fernanda Alzate Sánchez