

## EXAMEN DE CASOS APLICADOS – ALGORITMOS Y ESTRUCTURAS DE DATOS

**Duración:** 3 horas

**Lenguaje permitido:** C++ o Python

**Entrega:** código funcional

### Caso 1: Sistema de emergencia hospitalaria

Un hospital regional implementará un sistema para administrar la atención de pacientes según su gravedad. Cada paciente tiene: nombre, DNI, nivel de emergencia (1-5), y tiempo estimado de atención.

**Requisitos:**

1. Diseña un sistema que permita atender a los pacientes priorizando los casos más graves.
2. Permitir ver en tiempo real la lista de pacientes pendientes.
3. Mostrar cuántos pacientes fueron atendidos y su tiempo total de atención.

**Preguntas:**

- ¿Qué estructura(s) de datos usarías y por qué?
- ¿Qué algoritmo aplicarías para mantener el orden dinámico de prioridades?
- Implementa el código que simula la atención con entrada de datos.

### Caso 2: Análisis de rutas en una ciudad altiplánica

Una ciudad altiplánica con calles empinadas y rutas turísticas necesita un sistema que calcule si existe una ruta posible entre dos ubicaciones usando caminos existentes.

Se te proporciona un grafo no dirigido con nombres de puntos y las conexiones entre ellos.

**Requisitos:**

1. Implementar la entrada de datos para el grafo.
2. Verificar si existe camino entre dos puntos utilizando **DFS o BFS**.
3. Mostrar todas las rutas posibles de un punto a otro.

**Preguntas:**

- ¿Cuál estructura es más conveniente para representar el grafo?
- ¿Qué algoritmo conviene usar si se busca eficiencia?
- ¿Cómo optimizarías el recorrido?

### Caso 3: Sistema de historial de comandos en una aplicación de diseño

Una aplicación de diseño gráfico permite a los usuarios deshacer y rehacer sus acciones (dibujar, mover, eliminar objetos).

**Requisitos:**

1. Modela este sistema usando estructuras de datos.
2. Simula el uso de los comandos deshacer y rehacer.
3. Incluye límite máximo de 10 acciones almacenadas para optimizar memoria.

**Preguntas:**

- ¿Qué estructura de datos es ideal para este comportamiento?
- Justifica cómo se maneja la reversión de acciones con tus estructuras.
- Simula con un menú sencillo: [1] Dibujar, [2] Mover, [3] Eliminar, [4] Deshacer, [5] Rehacer.



### Evaluación (0 – 20 puntos)

Criterio	Puntaje Máximo
Análisis y elección de estructuras	6 puntos
Desarrollo y claridad del código	6 puntos
Funcionamiento correcto del programa	4 puntos
Explicación técnica y documentación	4 puntos
<b>Total</b>	<b>20 puntos</b>

**Nota:** Los estudiantes deberán desarrollar las soluciones de cada caso utilizando **interfaces gráficas simples** que permitan visualizar el comportamiento de las estructuras de datos implementadas (listas, pilas, colas, árboles o grafos). El uso de gráficos puede ser básico (por ejemplo, impresión visual por consola en forma de tablas o diagramas), pero debe representar claramente el flujo de los datos, las inserciones, eliminaciones, y/o recorridos. Se evaluará tanto la **funcionalidad del algoritmo** como la **capacidad de representar gráficamente la lógica estructural del problema**.

### Libros gratuitos sobre Estructuras de Datos y Algoritmos

1. Arriondo, R., Bottazzi, C., Costarelli, S., D'Elía, J., Dalcin, L., ... Yera Moreno, R. (2024). *Algoritmos y estructuras de datos* (versión 2.0.5). Universidad Nacional del Litoral. **Enlace:** <https://cimec.org.ar/~mstorti/aed/aednotes.pdf>
2. Bel, W. (2020). *Algoritmos y estructuras de datos en Python: Un enfoque ágil y estructurado*. Editorial UADER. **Enlace:** <https://editorial.uader.edu.ar/sites/default/files/imagenes-sitio/colecciones/amalgama/algoritmos-y-estructuras-de-datos-en-python/algoritmos-y-estructuras-de-datos-en-python-digital.pdf>
3. Córdor Tinoco, E. E., & De la Cruz Rocca, M. A. (2020). *Algoritmos resueltos con Python*. Editorial EIDEC. **Enlace:** <https://www.editorialeidec.com/wp-content/uploads/2020/10/Algoritmos-resueltos-con-Python.pdf>
4. Fridlender, D. (2018). *Apuntes para algoritmos y estructuras de datos II*. Universidad Nacional de Córdoba. **Enlace:** <https://www.famaf.unc.edu.ar/~gonzalez/ayed2/web/Apuntes/todo.pdf>

### Free Books on Data Structures and Algorithms (English)

5. Barnett, G., & Del Tongo, L. (2008). *Data structures and algorithms: Annotated reference with examples*. (Creative Commons ed.) DotNetSlackers. **Link:** <https://www.mta.ca/~rrosebru/oldcourse/263114/Dsa.pdf>



6. **Bible, P. W., & Moser, L. (2023).** *An open guide to data structures and algorithms.* PALNI Press. **Link:** <https://pressbooks.palni.org/anopenguidetodatastructuresandalgorithms/> (Texto abierto con descarga gratuita)
7. **Black, P. E. (Ed.). (2020).** *Dictionary of algorithms and data structures.* NIST. **Link:** <https://www.nist.gov/dads> (Referencia en línea de términos de EDA)
8. **Downey, A. B. (2015).** *Think Python: How to think like a computer scientist* (2nd ed.). O'Reilly Media. **Link:** <https://greenteapress.com/thinkpython2/thinkpython2.pdf>
9. **Downey, A. B. (2017).** *Think data structures: Algorithms and information retrieval in Java.* O'Reilly Media / Green Tea Press. **Link:** <http://greenteapress.com/thinkdast/thinkdast.pdf>
10. **Erickson, J. (2019).** *Algorithms.* (Self-published textbook). **Link:** <http://jeffe.cs.illinois.edu/teaching/algorithms/book/Algorithms-JeffE.pdf>
11. **Horvick, R. (2013a).** *Data Structures Succinctly, Part 1.* Syncfusion. **Link:** <https://freecomputerbooks.com/Data-Structures-Succinctly-Part1.html> (PDF gratuito disponible)
12. **Horvick, R. (2013b).** *Data Structures Succinctly, Part 2.* Syncfusion. **Link:** <https://freecomputerbooks.com/Data-Structures-Succinctly-Part2.html> (PDF gratuito disponible)
13. **Laaksonen, A. (2018).** *Competitive programmer's handbook.* (Draft ed.) **Link:** <https://cses.fi/book.html> (PDF gratuito del libro de 2018)
14. **Miller, B. N., & Ranum, D. L. (2013).** *Problem solving with algorithms and data structures using Python.* Franklin, Beedle & Associates. **Link:** <https://runestone.academy/runestone/books/published/pythonds/index.html>
15. **Morin, P. (2013).** *Open data structures: An introduction.* Athabasca University Press. **Link:** [https://www.aupress.ca/app/uploads/120226\\_99Z\\_Morin\\_2013-Open\\_Data\\_Structures.pdf](https://www.aupress.ca/app/uploads/120226_99Z_Morin_2013-Open_Data_Structures.pdf)
16. **Nievergelt, J., & Hinrichs, K. (2011).** *Algorithms and data structures: With applications to graphics and geometry.* Global Text Project (Jacobs Foundation). **Link:** [https://textbookequity.org/Textbooks/Nievergelt\\_Algorithms%20and%20Data%20Structures08.pdf](https://textbookequity.org/Textbooks/Nievergelt_Algorithms%20and%20Data%20Structures08.pdf)
17. **Sedgewick, R., & Wayne, K. (2011).** *Algorithms* (4th ed.). Addison-Wesley. **Link:** <https://algs4.cs.princeton.edu> (Sitio web del libro con material gratuito)
18. **Shaffer, C. A. (2012).** *Data structures & algorithm analysis in Java* (3rd ed.). Dover Publications (Open Access version). **Link:** <http://people.cs.vt.edu/shaffer/Book/JAVA3elatest.pdf>
19. **Shaffer, C. A. (2013).** *Data structures & algorithm analysis in C++* (3rd ed.). OpenDSA / Dover. **Link:** <http://people.cs.vt.edu/shaffer/Book/C++3elatest.pdf>



20. **Tóth, Š.** (2023). *A complete guide to standard C++ algorithms* (v1.0.0). (E-book autoeditado, beneficencia EFF). **Link:** [https://github.com/HappyCerberus/book-cpp-algorithms/releases/download/v1.0.0/book\\_with\\_cover.pdf](https://github.com/HappyCerberus/book-cpp-algorithms/releases/download/v1.0.0/book_with_cover.pdf)
21. **Wirth, N.** (2012). *Algorithms and data structures* (Oberon version). ETH Zurich. **Link:** <http://www.ethoberon.ethz.ch/WirthPubl/AD.pdf>

