

REQUERIMEINTOS	HISTORIA DE USUARIO
RF01 – Validar datos de entrada Validar que los datos de los estudiantes ingresados sean correctos y que cada estudiante tenga una nota válida.	Validar los datos ingresados de los estudiantes para asegurar que las notas sean valores numéricos y permitan realizar el cálculo del promedio sin errores.
RF02 – Registrar estudiantes Registrar la información de los estudiantes, incluyendo nombre y nota, para su posterior procesamiento.	Registrar los estudiantes con su respectivo nombre y nota para que el sistema pueda almacenar y procesar la información académica.
RF03 – Construir tabla de programación dinámica Construir una tabla de programación dinámica de tamaño $n \times n$ para almacenar las sumas parciales de las notas.	Construir una tabla de programación dinámica que permita almacenar las sumas parciales de las notas y evitar cálculos repetidos.
RF04 – Calcular promedio del grupo Calcular el promedio general de los estudiantes utilizando la tabla de programación dinámica.	Calcular el promedio del grupo de estudiantes a partir de los valores almacenados en la tabla de programación dinámica.
RF05 – Identificar estudiantes aprobados Identificar los estudiantes cuya nota sea mayor o igual a 3.0.	Identificar los estudiantes aprobados para poder mostrar únicamente aquellos que cumplen con el criterio mínimo de aprobación.

REQUERIMIENTOS NO FUNDAMENTALES	
---------------------------------	--

RNF01 – Eficiencia del algoritmo Requerimiento Garantizar que el algoritmo tenga una complejidad temporal aceptable.	
RNF02 – Uso de memoria Requerimiento Controlar el uso de memoria durante la ejecución del sistema.	
RNF03 – Claridad de resultados Presentar los resultados de forma clara y entendible.	

Promedio de notas:

COST	TIMES
C1	1
C2	1
C3	$n^2$
C4	1
C5	n
C6	n
C7	1
C8	$n^2/2$
C9	$n^2/2$

C10	1
-----	---

$$T(n) = C3n^2 + C8n^2 + C9n^2 + (C5 + C6 + C7)n + (C1 + C2 + C4 + C10)$$

$$T(n) = C11n^2 + C12n + C13$$

$$\text{Complejidad} = O(n^2)$$

Estudiantes aprobados:

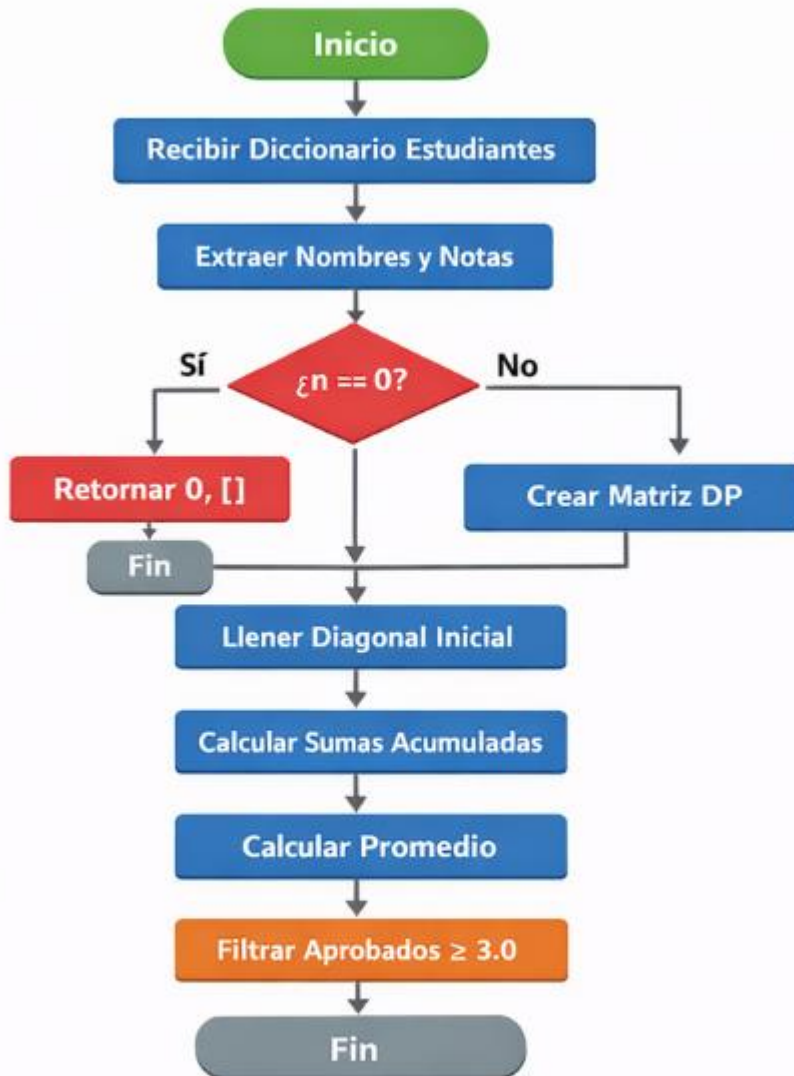
COST	TIME
C1	n
C2	n
C3	1
C4	$n^2$
C5	n
C6	$n^2$
C7	n
C8	1

$$T(n) = (C4 + C6)n^2 + (C1 + C2 + C5 + C7)n + (C3 + C8)$$

$$T(n) = C_9n^2 + C_{10}n + C_{11}$$

Complejidad de  $\Theta(n^2)$

### Algoritmo: Promedio y Aprobados con DP



## Secuencia de Ejecución

