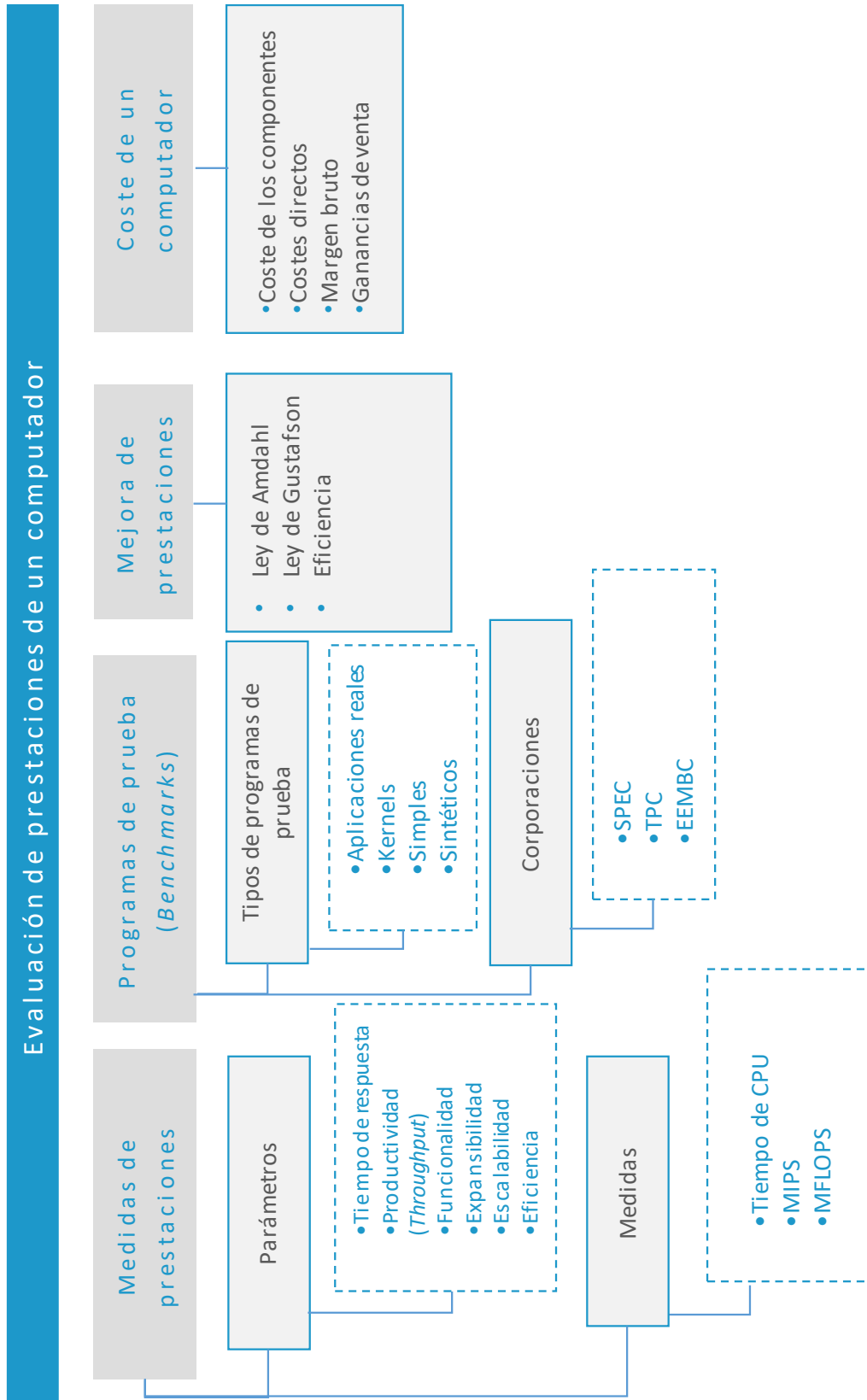


Estructura de Computadores

Evaluación de prestaciones de un computador

Índice

Esquema	3
Ideas clave	4
2.1. Introducción y objetivos	4
2.2. Medidas de prestaciones	5
2.3. Programas de prueba (<i>benchmarks</i>)	9
2.4. Mejora de prestaciones	10
2.5. Coste de un computador	13
2. 6. Referencias bibliográficas	13
A fondo	14
Test	16



2.1. Introducción y objetivos

Para estudiar este tema lee las **ideas clave** que se desarrollan a continuación.

En este tema presentaremos algunos conceptos generales de la evaluación de prestaciones de un computador, sin perjuicio de que en el desarrollo posterior de la asignatura volvamos sobre estos conceptos para aplicarlos a casos concretos.

Empezaremos presentando algunos parámetros típicos de la **medida de prestaciones**, como el **Tiempo de CPU**, las **MIPS** o los **MFLOPS**.

A continuación, comentaremos algunas de las aplicaciones que se emplean o se han empleado históricamente como estándares de referencia (**benchmarks**) en la comparación de computadores.

Posteriormente observaremos cómo evaluar las mejoras que se pueden implementar en una configuración, aún en modo parcial, empleando, para valorar su efecto, algunas expresiones matemáticas, como la **ley de Amdahl**, entre otras.

Por último, se presentarán algunos aspectos que caracterizan el precio final de un computador.

2.2. Medidas de prestaciones

Existen distintos **parámetros** que permiten evaluar las prestaciones de un computador (Ortega et al., 2005):

- ▶ Tiempo de respuesta.
- ▶ Productividad (*Throughput*).
- ▶ Funcionalidad.
- ▶ Expansibilidad.
- ▶ Escalabilidad.
- ▶ Eficiencia.

Cada uno de estos parámetros afectará de distinta forma las prestaciones e incluso se denominará de manera diferente en función del elemento del computador a estudiar. Ejemplos:

Elemento	Entradas	Medidas más habituales
Organización de memoria	Accesos a memoria	<ul style="list-style-type: none">▶ Tiempo de latencia▶ Ancho de banda▶ Escalabilidad
Computador	Programas	<ul style="list-style-type: none">▶ Tiempo de respuesta▶ Productividad▶ Funcionalidad
Procesador	Instrucciones	<ul style="list-style-type: none">▶ CPI, CPI▶ Frecuencia de reloj o duración de ciclo▶ Nº de instrucciones que procesa

Tabla 1. Ejemplos de entradas y medidas más habituales en distintos elementos de un computador.

En el caso de un procesador se emplean medidas como el **Tiempo de CPU** (T_{CPU}), para evaluar la ventaja de aplicar determinadas técnicas de paralelismo a una configuración, mejorando así su productividad (instrucciones procesadas por unidad de tiempo) o los **MIPS** (Millones de Instrucciones por Segundo) o los **MFLOPS**

(Millones de Operaciones en Coma Flotante por Segundo), que pretenden facilitar la comparación de prestaciones entre computadores, sobre un código de referencia.

Tiempo de CPU de un programa

Es el tiempo requerido para que una CPU determinada ejecute un programa y es el inverso del **rendimiento**. Se evalúa mediante la expresión:

$$T_{CPU} = NI \times CPI \times T_{CICLO} = NI \times \left(\frac{CPI}{f} \right)$$

[2.1]

Donde:

- ▶ **NI**: es el número de instrucciones ejecutadas del programa o fragmento de programa.
- ▶ **T_{ciclo}**: Periodo de la señal de reloj del procesador, duración de un ciclo e inverso de la frecuencia (**f**) del reloj de la CPU.
- ▶ **CPI**: Número medio de ciclos por instrucción.

Si todas las instrucciones duran el mismo número de ciclos, CPI coincidirá con el cociente entre el número de ciclos consumidos por el programa y el número de instrucciones ejecutadas.

Lo habitual es que las instrucciones tengan distinta duración, en este caso, suponiendo que hay n tipos distintos de instrucciones:

$$CPI = \frac{\sum_{i=1}^n NI_i \times CPI_i}{NI}$$

[2.2]

Donde:

- ▶ **NI_i**: es el número de instrucciones de tipo i-ésimo que existen en el programa o fragmento de programa.
- ▶ **CPI_i**: Duración en ciclos de la ejecución de una instrucción de tipo i-ésimo.

Para optimizar un computador se busca reducir el T_{CPU} , para ello se pueden intentar reducir cada uno de los tres factores que aparecen en la expresión [2.1]. La tabla que aparece a continuación indica la dependencia que tienen cada uno de esos factores de distintos aspectos del computador, o relacionados con el mismo:

	Tecnología	Estructura y Organización	Conjunto de instrucciones	Compilador
NI			X	X
CPI	X	X	X	X
Tciclo	X	X		

Tabla 2. Dependencia de los factores del T_{CPU} de distintos aspectos del computador.

Ortega, Anguita y Prieto (2005) indican algunos ejemplos sobre cómo la actuación sobre esos aspectos proporciona diferentes soluciones de optimización y tipos de arquitecturas paralelas, compatibles entre sí en muchos casos.

MIPS

Se trata de una **medida de rendimiento (inverso de tiempo)** que pretende agrupar el tiempo de respuesta, la productividad y la funcionalidad en un único parámetro, se le designa como un tiempo de respuesta para una entrada compleja (Ortega et al., 2005).

Para ello se emplean los MIPS (Millones de Instrucciones Por Segundo), que, en función del T_{CPU} se calcula como:

$$\text{MIPS} = \frac{\text{NI}}{T_{\text{CPU}} \times 10^6} = \frac{f}{\text{CPI} \times 10^6}$$

[2.3]

Esta medida depende del programa ejecutado y, por lo tanto, del conjunto de instrucciones y se emplea para comparar **máquinas con conjunto de instrucciones compatibles**, mediante programas que se establecen como referencia.

Los MIPS son una medida de la velocidad de ejecución de instrucciones (rendimiento).

MFLOPS

Los MFLOPS (*Mega Floating point Operations Per Second*, Millones de Operaciones en Coma Flotante Por Segundo), son una medida de rendimiento de características similares a los MIPS pero mucho más específica, puesto que solo evalúa las operaciones en coma flotante realizadas.

Se calculan mediante la siguiente expresión:

$$\text{MFLOPS} = \frac{\text{Operaciones_en_Coma_Flotante}}{T_{\text{CPU}} \times 10^6}$$

[2.4]

El conjunto de operaciones en coma flotante no es constante en máquinas diferentes y la potencia de las operaciones en coma flotante no es igual para todas las operaciones (por ejemplo, con diferente precisión, no es igual una suma que una multiplicación). Para paliar este problema se emplean en ocasiones **MFLOPS**

normalizados, que se calculan dando un peso relativo a cada instrucción en función de las que se consideren básicas.

$$\text{MFLOPS}_{\text{normalizados}} = \frac{\sum_{i=1}^n \text{Operaciones_en_Coma_Flotante}_i \times W_i}{T_{\text{CPU}} \times 10^6}$$

[2.5]

Donde se consideran n tipos de instrucciones en coma flotante diferentes y W_i es el peso o coste de la instrucción i -ésima en relación con la menos costosa (Ortega et al., 2005).

2.3. Programas de prueba (*benchmarks*)

Para comparar computadores de forma fiable, lo más adecuado es evaluar sus prestaciones cuando ejecutan un conjunto de **programas de prueba (*benchmarks*)** seleccionados de forma que representen la carga de trabajo usual de las máquinas a evaluar, ejercitando los distintos elementos del computador de forma similar.

Tipos de programas de prueba (Ortega et al., 2005):

- ▶ **Aplicaciones reales** (Compiladores de C, Word, Photoshop...). Pueden presentar problemas de portabilidad relacionados con la dependencia del compilador o del S.O. A veces se utilizan aplicaciones modificadas (*scripted applications*) para simular interacciones multiusuario complejas (servidores), o quitar operaciones de I/O para medir bien el comportamiento de la CPU.
- ▶ **Kernels** (Bucles de Livermoore, Linpacks). Pequeños trozos de programas reales seleccionados para evaluar características específicas de una máquina o explicar las causas de las diferencias entre máquinas distintas.

- ▶ **Simples** (Toys) (Criba de Eratóstenes, Puzzle, Quicksort). Programas pequeños (10-100 líneas), fáciles de escribir, y de resultado conocido.
- ▶ **Sintéticos** (Dhrystone, Whetstone). Programas que reproducen los porcentajes de instrucciones y uso de recursos de cargas de trabajo reales.

Existen instituciones o agrupaciones que tratan de normalizar los conjuntos de programas de prueba, para ofrecer a fabricantes y clientes herramientas de comparación homologadas. Entre estos consorcios destacan:

- ▶ **SPEC** (Standard Performance Evaluation Corporation). Probablemente la más conocida, dedicada fundamentalmente a servidores y computadores personales.
- ▶ **TPC** (Transaction Processing Council). Evaluación de servidores y sistemas de que corren bases de datos.
- ▶ **EEMBC** (EDN Embedded Microprocessor Benchmark Consortium). Para computadores embebidos o empotrados (microcontroladores).

2.4. Mejora de prestaciones

Si en un computador se **mejoran** las prestaciones de un componente o elemento haciendo que su **velocidad** (rendimiento) sea p veces mayor (ejemplos: se utilizan p procesadores en lugar de uno, la ALU realiza las operaciones en un tiempo p veces menor). El incremento de velocidad que se consigue en la nueva situación con respecto a la previa (**máquina base o de referencia**) y se expresa mediante la **ganancia de velocidad, aceleración o *speed—up*** (S_p).

$$S_p = \frac{V_p}{V_l} = \frac{T_l}{T_p}$$

[2.6]

Donde:

- ▶ V_1 : Velocidad (o prestaciones, en general) de la máquina base.
- ▶ V_p : Velocidad de la máquina mejorada **p** veces.
- ▶ T_1 : Tiempo de respuesta de la máquina base.
- ▶ T_p : Tiempo de respuesta de la máquina mejorada **p** veces.

Ley de Amdahl

Calcula la ganancia de velocidad de un computador en el que se introduce una mejora pero que no se puede beneficiar la aplicación ejecutada completa:

$$S_p = \frac{T_1}{T_p} \leq \frac{p}{1 + f \times (p - 1)}$$

[2.7]

Donde:

- ▶ **f**: es la fracción de tiempo de ejecución (o de código) sin mejora.
- ▶ **p**: mejora parcial introducida.

Amdahl la enunció en relación con la eficacia de los computadores paralelos, dado que en un programa hay código secuencial que no puede paralelizarse y los procesadores no se podrían utilizar eficazmente:

$$T_p \geq f \times T_1 + (1 - f) \times \frac{T_1}{p}$$

[2.8]

Es más efectivo mejorar poco lo que más se usa que mejorar mucho lo que menos se usa.

Según la **ley de Amdahl**, la aceleración está limitada por f (figura 2.1) porque siempre opera con la **misma carga de trabajo** (misma tarea). Está considerada como una ley pesimista. Existen otros modelos, como la **ley de Gustafson**, que palian esa limitación.

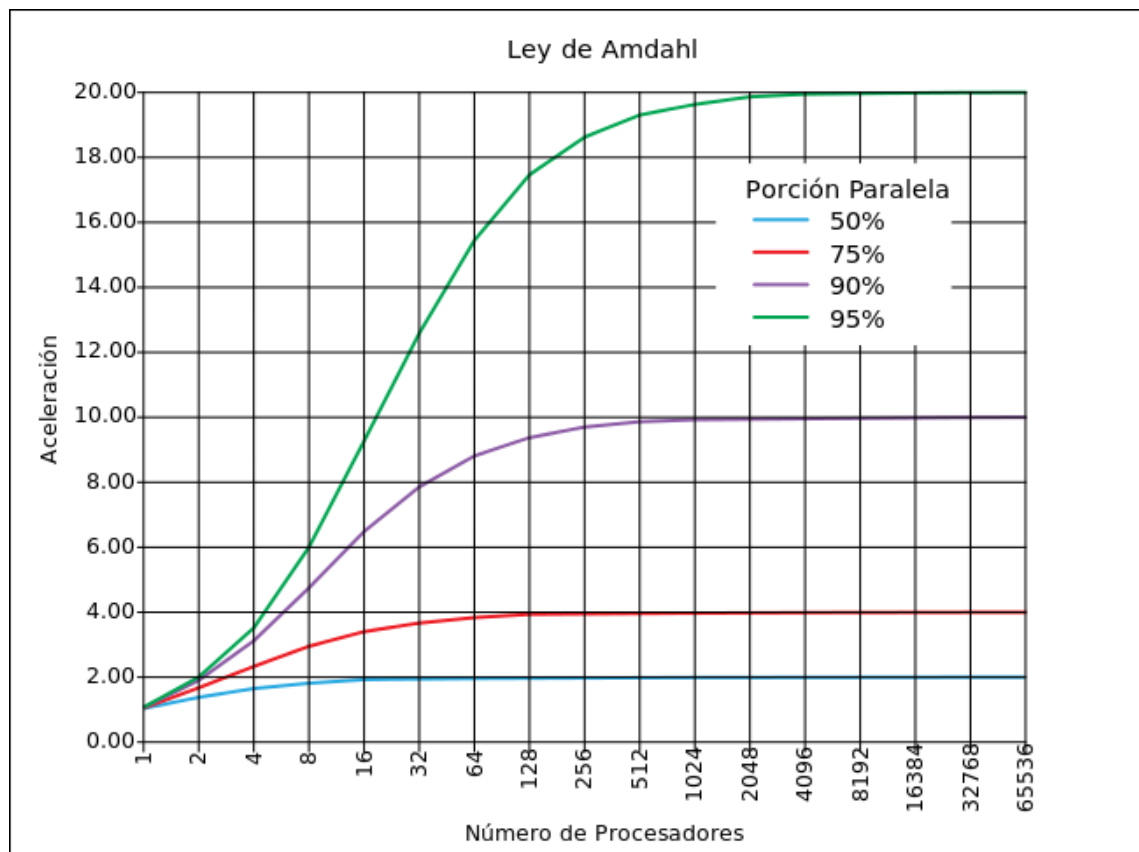


Figura 1. Representación gráfica de la ley de Amdahl en función de f .

Fuente: <http://commons.wikimedia.org/wiki/File:AmdahlsLaw.svg>

La eficiencia **evalúa** en qué medida se aprovechan los recursos del sistema, dicho de otra forma, si las prestaciones obtenidas se acercan a las máximas obtenibles.

2.5. Coste de un computador

En este apartado se presentan distintos aspectos que fijan el **precio** de un computador, relacionándolo con su **coste**.

El coste de los componentes de un computador decrece con el tiempo (aunque no haya mejoras importantes en la tecnología). Ello es debido a la llamada **curva de aprendizaje** (*learning curve*) que ocasiona mejoras en el rendimiento de los procesos de fabricación (si se dobla el rendimiento de un proceso se consigue reducir a la mitad el coste).

La siguiente figura muestra los factores que determinan el precio final de un computador (Ortega et al., 2005).

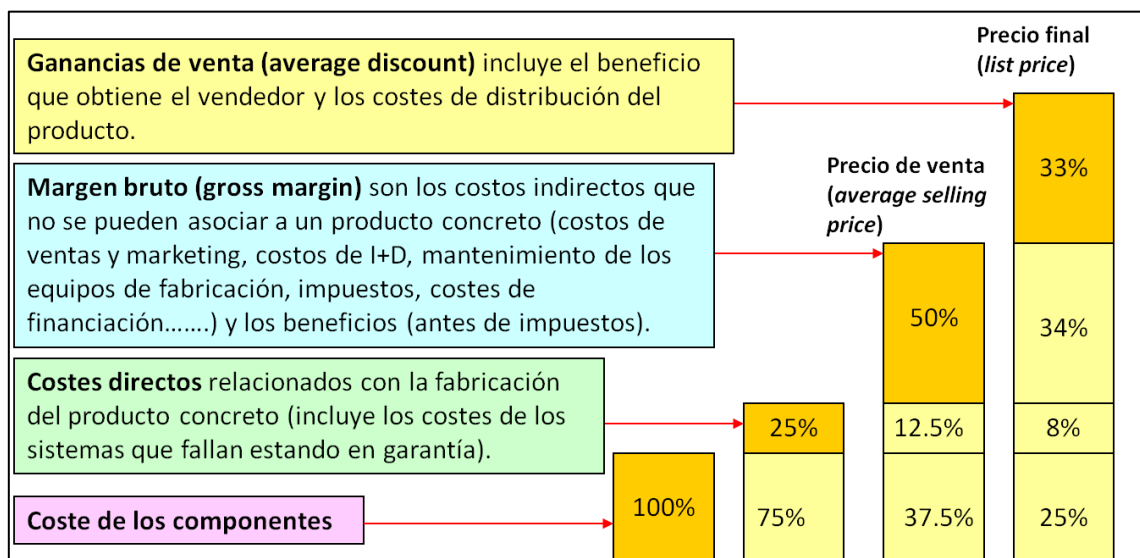


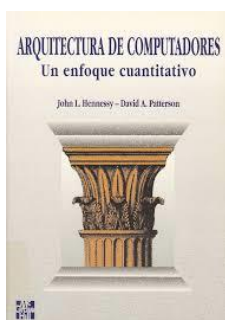
Figura 2. Factores que determinan el precio final de un computador. Fuente: Ortega et al., 2005.

2. 6. Referencias bibliográficas

Ortega, J., Anguita, M. y Prieto, A. (2005). *Arquitectura de computadores*. Madrid: Thomson.

Rendimiento y coste

Hennessy, J. y Patterson, D. (2002). *Arquitectura de computadores. Un enfoque cuantitativo*. Madrid: McGraw-Hill.



Para ampliar algunos aspectos relacionados con los factores de evolución del computador es interesante revisar el segundo capítulo: Rendimiento y coste (páginas 35-74).

Accede al documento a través del aula virtual

Las 500 supercomputadores más poderosas del mundo

En esta página se establece una clasificación de los 500 supercomputadores más potentes según el benchmark Linpack.



Accede a la página a través del aula virtual o desde la siguiente dirección web:

<http://www.top500.org/>

Standard Performance Evaluation Corporation (SPEC)

Standard Performance Evaluation Corporation (SPEC). Página web oficial.
<http://www.spec.org/>



Esta es la página del consorcio SPEC que se dedica a evaluar servidores y ordenadores personales.

Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities

Amdahl, G. (1967). Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. *AFIPS Conference Proceedings*, 30, 483-485. Recuperado de <http://www-inst.eecs.berkeley.edu/~n252/paper/Amdahl.pdf>

Buscando ir a las fuentes originales, es interesante revisar el artículo en el que Gene Amdahl enunció su mencionada ley.

Revisión de las métricas de evaluación en arquitecturas de computadores. Ley de Gustafson

Gustafson, J. (1988). Reevaluating Amdahl's Law. *Communications of the ACM*, 31. Recuperado de <http://www.johngustafson.net/pubs/pub13/amdahl.htm>

Como ya se ha comentado anteriormente, la ley de Gustafson aporta una evaluación más «optimista» del efecto de las mejoras en un computador, como describe el propio autor en su artículo original.

1. Respecto a las prestaciones de un procesador:
 - A. La escalabilidad plantea la posibilidad de ampliar la capacidad de procesamiento añadiendo bloques a la arquitectura existente.
 - B. La expansibilidad plantea la posibilidad de ampliar el sistema sin que se devalúen las prestaciones.
 - C. La productividad es la relación entre las prestaciones obtenidas y el coste que ha supuesto conseguir las.
 - D. Ninguna de las tres respuestas anteriores es correcta.

2. El tiempo de CPU de un programa...
 - A. Es inversamente proporcional al número medio de ciclos que requiere una instrucción.
 - B. Es directamente proporcional al periodo de la señal de reloj del procesador.
 - C. Es independiente del número de instrucciones procesadas.
 - D. Ninguna de las tres respuestas anteriores es correcta.

3. Respecto a la ejecución de instrucciones de un procesador:
 - A. Se puede decir que, en general, todas las instrucciones requieren el mismo número de ciclos para ser ejecutadas.
 - B. Todas las instrucciones se ejecutan una sola vez.
 - C. El número de instrucciones ejecutadas depende de la tecnología del procesador.
 - D. Las tres respuestas anteriores son falsas.

4. Se dispone de un computador basado en un Microprocesador a 2GHz. Se ejecuta un programa compuesto por 1000 instrucciones. La tabla muestra la proporción de instrucciones que integran el programa y la duración de cada tipo en ciclos.

Tipo de instrucción	Instrucciones/Prog.	CPIi
Lógico—Aritmética	20 %	4
Salto incondicional	10 %	3
Salto condicional	10 % / 10 %	7/5
Carga (Reg<-Mem)	30 %	3
Almacenamiento (Mem<-Reg)	20 %	4

¿Cuánto vale el CPI resultante?

- A. 5 ciclos.
 - B. 6 ciclos.
 - C. 4 ciclos.
 - D. 3 ciclos.
5. En las condiciones de la pregunta anterior, ¿cuál es el tiempo de CPU del programa?:
- A. 2000 μ s.
 - B. 1000 ns.
 - C. 3000 μ s.
 - D. 2000 ns.
6. En un computador se sustituye la ALU por una nueva diez veces más rápida, sabiendo que el programa que se va a ejecutar usa la ALU un 40 % del tiempo que dura el proceso ¿qué ganancia o aceleración (S) se ha obtenido con el cambio?
- A. 3'25.
 - B. 1'56.
 - C. 0'57.
 - D. 2'34.

7. Respecto a MIPS y MFLOPS se puede decir:

- A. Ambas son medidas de eficiencia.
- B. Ambas son aplicables a cualquier tipo de computador.
- C. Ambas miden velocidad de proceso.
- D. Las tres respuestas anteriores son correctas.

8. Los *benchmarks*...

- A. Son aplicaciones de prueba que se emplean para comparar computadores de características compatibles.
- B. Son elementos hardware que miden el rendimiento de un computador.
- C. Son aplicaciones que permiten corregir errores de proceso en un computador.
- D. Son un tipo de computador patrón.

9. Con respecto a la ley de Amdahl:

- A. Sirve para calcular la eficiencia de un computador.
- B. Se usa para identificar cuántas instrucciones se procesan por unidad de tiempo.
- C. Evalúa el tiempo necesario para finalizar un programa.
- D. Ninguna de las respuestas anteriores es correcta.

10. Respecto a las medidas de mejora en una configuración:

- A. La ley de Amdahl es aplicable en todos los casos y no presenta ningún tipo de limitación.
- B. La ley de Gustafson es una medida de mejora que no trabaja con carga de trabajo fija.
- C. La ganancia según la ley de Gustafson es inversamente proporcional a la mejora introducida.
- D. Ninguna de las afirmaciones anteriores es correcta.