



La Automatización del Ajedrez DeepBlue

Historia de la Computación

Daniel Machado Pérez, C411

Ariel González Gómez, C412

9 de mayo de 2025



Universidad de La Habana
Facultad de Matemática y Computación

Repositorio en GitHub

Índice

1. Introducción	4
2. Antecedentes del ajedrez computacional	5
2.1. De ChipTest a Deep Thought	5
2.2. ChipTest	5
2.3. Primeros éxitos y limitaciones	6
3. Origen y maduración de Deep Blue	6
3.1. Deep Blue I (1995-1996)	6
3.2. Deep Blue II (1996-1997)	7
4. Arquitectura del sistema	7
4.1. Infraestructura IBM RS/6000 SP	8
4.2. Chips especializados	9
4.3. Jerarquía Maestro-Trabajador-Chip	9
5. Algoritmos de búsqueda y evaluación	10
5.1. Búsqueda selectiva híbrida	10
5.2. Evaluación en <i>hardware</i>	10
5.3. Extensiones y búsqueda quiescente	11
6. Paralelismo y rendimiento	11
6.1. Paralelismo en <i>software</i>	11
6.2. <i>Offloading</i> a <i>hardware</i>	12
6.3. Medición de eficiencia	12
7. Componentes de apoyo	12
7.1. Libro de aperturas	12
7.2. <i>Extended Book</i>	13
7.3. Bases de datos de finales	13
7.4. Control de tiempo	13
8. Impacto y legado	14
8.1. Influencias posteriores	14
8.2. Repercusiones histórico-culturales	15
8.3. Curiosidades	15
9. Conclusiones	16

Resumen

El presente informe analiza en detalle la evolución y los componentes técnicos de *Deep Blue*, la primera máquina capaz de derrotar al campeón mundial de ajedrez bajo condiciones reglamentarias. Se describen los antecedentes desde ChipTest y Deep Thought, la arquitectura de *hardware* VLSI y del clúster IBM RS/6000 SP, los algoritmos de búsqueda híbrida en C con créditos diferidos y la función de evaluación implementada en silicio. Además, se examinan los mecanismos de paralelismo y balance de carga, el soporte de libros de aperturas y bases de finales, así como las estrategias de control de tiempo. Finalmente, se evalúa el impacto histórico y el legado de *Deep Blue* en la supercomputación y la inteligencia artificial aplicada, destacando su papel en el paso de enfoques de fuerza bruta a sistemas híbridos con conocimiento experto.

Palabras clave

Deep Blue, ajedrez computacional, arquitectura paralela, IBM RS/6000 SP, búsqueda selectiva, chip VLSI, supercomputación.

1 Introducción

Desde sus inicios, el juego de ajedrez ha sido considerado un terreno fértil para evaluar la capacidad de las máquinas para replicar formas avanzadas de pensamiento humano. La complejidad combinatoria del ajedrez, junto con su estructura formal y reglas bien definidas, lo convierten en un banco de pruebas ideal para los sistemas de inteligencia artificial. En este contexto, el desarrollo de *Deep Blue* por parte de IBM representa un hito en la historia de la computación, no solo por su impacto mediático, sino también por los avances técnicos y conceptuales que introdujo en la automatización del razonamiento estratégico.

Deep Blue no fue un simple programa de ajedrez, sino un sistema computacional altamente especializado que integraba componentes de *hardware* diseñados específicamente para el cálculo masivo de posiciones y movimientos posibles, junto con sofisticados algoritmos de búsqueda y evaluación posicional. La arquitectura del sistema combinó miles de procesadores en paralelo con chips dedicados a operaciones de ajedrez, permitiendo una profundidad de búsqueda y una velocidad de procesamiento inalcanzables para cualquier computadora convencional de su época.

Más allá de su capacidad para vencer a un campeón mundial, lo que convierte a *Deep Blue* en un objeto de estudio técnico relevante es su enfoque híbrido de solución: una articulación precisa entre potencia de cálculo, paralelismo masivo y conocimiento experto codificado. Este informe se centra en analizar los componentes técnicos que hicieron posible dicha hazaña, incluyendo tanto los aspectos de *hardware* como los mecanismos algorítmicos empleados, y cómo estos interactúan de forma coordinada para lograr un rendimiento sobresaliente en la resolución de un problema complejo como el ajedrez.

El objetivo de este estudio no es explorar los aspectos históricos del enfrentamiento entre *Deep Blue* y Garry Kasparov, aunque se comentará brevemente su contexto; el foco principal recae sobre los elementos estructurales y funcionales del sistema, desde su arquitectura paralela hasta las heurísticas que guían la evaluación de posiciones. A través de este análisis técnico, se busca comprender cómo la integración eficaz de recursos computacionales puede acercarse, e incluso superar, ciertas formas de cognición humana en dominios bien estructurados.

2 Antecedentes del ajedrez computacional

Entre finales de la década de 1940 y principios de la de 1950, surgieron los primeros esfuerzos teóricos y conceptuales para dotar a las máquinas de la capacidad de jugar al ajedrez. En 1949, Claude E. Shannon publicó el artículo “*Programming a Computer for Playing Chess*”, en el que planteó por primera vez el uso de la técnica minimax combinada con funciones de evaluación heurísticas y un corte alfa-beta rudimentario para reducir el espacio de búsqueda [15]. En este trabajo, Shannon (considerado el padre de la teoría de la información) estimó la complejidad del árbol de juego de ajedrez en el orden de 10^{120} posiciones y sugirió métodos sistemáticos para explorar únicamente las ramas más prometedoras, estableciendo así las bases de la mayoría de los programas de ajedrez posteriores.

Pocos años después, en 1950-1952, Alan M. Turing y David G. Champernowne desarrollaron “Turochamp”, el primer programa de ajedrez propiamente dicho, concebido para ejecutarse en el ordenador ACE de la National Physical Laboratory (aunque nunca llegó a implementarse íntegramente en máquina y fue simulado paso a paso por Turing en papel) [17]. A pesar de su simplicidad y de que un solo movimiento requería más de media hora de cálculo humano, Turochamp demostró que incluso un algoritmo basado en reglas limitadas era capaz de jugar una partida reconocible, articulando con ello el paradigma de combinar heurísticas de posición, generación de jugadas y cortes selectivos, prefigurando los sistemas que conformarían la era moderna del ajedrez computacional.

2.1 De ChipTest a Deep Thought

En la década de los 80 se manifestaban esfuerzos tempranos en el diseño de una máquina capaz de jugar ajedrez a un nivel comparable al de los grandes maestros. Fue así como en la Universidad Carnegie Mellon nacían los proyectos ChipTest y Deep Thought, precursores de Deep Blue.

2.2 ChipTest

El proyecto ChipTest se inició en 1985 bajo la dirección de Feng-hsiung Hsu, Thomas Anantharaman y Murray Campbell, con el objetivo de explorar la viabilidad de un generador de movimientos de ajedrez implementado en VLSI (*Very Large Scale Integration*). La primera versión de ChipTest estaba controlada por una estación de trabajo Sun-3/160¹ y alcanzaba aproximadamente 50000 posiciones por segundo. En 1986, ChipTest compitió en el North American Computer Chess Championship (NACCC)², donde, pese a contar con pruebas limitadas antes del torneo, consiguió recuperarse tras un inicio adverso (3 derrotas) y obtuvo un empate [7].

En agosto de 1987 se rediseñó ChipTest y se rebautizó como ChipTest-M, multiplicando por diez su velocidad de búsqueda hasta alrededor de 500000 posiciones

¹Sun-3 es una serie de *computer workstations* y servidores de UNIX producida por Sun Microsystems, lanzada el 9 de septiembre de 1985.

²Campeonato de ajedrez computacional que se celebró entre 1970 y 1994. Organizado por la It was organised by the Association for Computing Machinery (ACM) y por Monty Newborn, profesor de Ciencia de la Computación en la Universidad McGill.

por segundo, ya sobre una Sun-4³. Esta versión se alzó con el título en el NACCC de 1987 con un marcador de 4-0. El éxito de ChipTest-M condujo al desarrollo de Deep Thought: la versión 0.01 apareció en mayo de 1988 y la 0.02 en noviembre del mismo año, incorporando dos procesadores VLSI dedicados y elevando la tasa de búsqueda a casi 700000 posiciones por segundo. Con Deep Thought, el equipo obtuvo el título en el World Computer Chess Championship de 1989 con un resultado perfecto de 5-0 [14].

2.3 Primeros éxitos y limitaciones

Deep Thought destacó en 1988 al compartir el primer puesto en el Software Toolworks Open de Los Ángeles⁴, donde venció al gran maestro Bent Larsen, consiguiendo una valoración de rendimiento de 2551 según la USCF (*US Chess Federation*) y ganando el Fredkin Intermediate Prize [9]. Asimismo, ganó el ACM Computer Chess Championship en Orlando ese mismo año [9]. Pese a estos logros, la arquitectura de ChipTest y Deep Thought permanecía limitada por un grado de paralelismo moderado, lo que restringía la profundidad uniforme de búsqueda y obligaba a depender intensamente de heurísticas de selección y extensiones locales para obtener rendimiento competitivo [7].

3 Origen y maduración de Deep Blue

A finales de 1989, parte del equipo original de ChipTest y Deep Thought (Feng-hsiung Hsu y Murray Campbell) pasó al IBM T.J. Watson Research Center para continuar el desarrollo de máquinas de ajedrez de alto rendimiento. En este nuevo entorno se estableció el proyecto *Deep Thought 2*, considerado el paso intermedio hacia lo que pronto sería conocido como *Deep Blue* [4]. Bajo la dirección de la división de investigación de IBM, el sistema heredó la arquitectura basada en chips VLSI especializada para generar y evaluar movimientos de ajedrez, al tiempo que incorporaba herramientas de *software* más avanzadas para depuración, afinamiento de la función de evaluación y análisis de partidas [4].

3.1 Deep Blue I (1995-1996)

El prototipo *Deep Blue I* comenzó a tomar forma en 1995, tras tres años de diseño de un chip de búsqueda único. La primera remesa de circuitos llegó en septiembre de ese año y, una vez solucionados los defectos iniciales, en enero de 1996 se dispuso de la versión estable. El sistema corría sobre un clúster IBM RS/6000 SP⁵ de 36 nodos y empleaba 216 chips VLSI, cada uno capaz de explorar entre 1.6 y 2 millones de posiciones por segundo. En conjunto, la máquina alcanzaba velocidades de entre 50 y 100 millones de posiciones por segundo [4].

³Aparecida por primera vez en julio de 1987, con el lanzamiento de la Sun 4/260.

⁴Torneo de ajedrez celebrado en 1988 en Los Ángeles, notable por su relevancia en la historia del ajedrez computacional

⁵Scalable POWERparallel es una serie discontinuada de supercomputadoras de la línea RS/6000 de IBM



Figura 1: Equipo central de Deep Blue, Joe Hoane, Feng-hsiung Hsu y Murray Campbell.

En febrero de 1996, *Deep Blue I* se enfrentó a Garry Kasparov bajo condiciones de torneo: tras empatar 2-2 en las cuatro primeras partidas, Kasparov ganó los dos encuentros finales, venciendo por 4-2 [4]. Aunque el resultado favoreció al campeón humano, la serie demostró que una máquina especialmente diseñada podía competir de tú a tú con la élite mundial.

3.2 Deep Blue II (1996-1997)

Tras el match de 1996 quedó patente la necesidad de mejorar la función de evaluación y aumentar el paralelismo. En respuesta, IBM diseñó un nuevo chip que amplió los rasgos de evaluación de alrededor de 6400 a más de 8000, incorporó detección de repeticiones en *hardware* y optimizaciones en generación de jugadas, elevando la velocidad de cada chip a 2-2.5 millones de posiciones por segundo. Además, el sistema pasó a disponer de 480 chips VLSI distribuidos en 30 nodos SP, duplicando con creces la capacidad de búsqueda [4].

Entre enero y mayo de 1997, el equipo dedicó la mayor parte de sus esfuerzos a afinar la función de evaluación y a preparar la base de datos de aperturas y finales. El resultado fue *Deep Blue II*, que derrotó a Kasparov por $3\frac{1}{2}$ - $2\frac{1}{2}$ en mayo de 1997, convirtiéndose en la primera máquina en vencer al campeón mundial en un match reglamentario y recibiendo por ello el Premio Fredkin⁶ [4].

4 Arquitectura del sistema

La solidez de la capacidad de cómputo de *Deep Blue* descansa en su diseño arquitectónico, que combina un clúster de propósito general de alto rendimiento con cientos de aceleradores VLSI especializados. Esta configuración híbrida permite delegar las primeras fases de la búsqueda en un entorno distribuido de nodos IBM

⁶Premio creado por Edward Frenkin para la primera computadora que venciera a un campeón mundial de ajedrez.



Figura 2: Deep Blue.

RS/6000 SP, optimizado para la gestión de tablas de transposición y la coordinación de tareas, y delegar las fases más profundas y tácticamente críticas en chips dedicados que ejecutan en silicio algoritmos de evaluación y control de búsqueda con latencia mínima [4]. A continuación se presentan los principales componentes de esta arquitectura, desde la infraestructura de los nodos hasta la jerarquía maestro-trabajador-chip que maximiza el paralelismo y la eficiencia del sistema.

4.1 Infraestructura IBM RS/6000 SP

El corazón de *Deep Blue* es un clúster IBM RS/6000 SP compuesto por treinta nodos equipados con procesadores Power2 Super Chip (P2SC)⁷, de los cuales veintiocho funcionan a una frecuencia de 120 MHz y dos a 135 MHz. Cada nodo dispone de 1 GB de memoria RAM y 4 GB de almacenamiento local. La interconexión se realiza mediante un conmutador de alta velocidad que implementa el estándar MPI (*Message-Passing Interface Standard*), garantizando una latencia mínima en el envío

⁷Microprocesadores diseñados y comercializados por IBM para servidores y supercomputadoras

y recepción de posiciones de ajedrez durante la búsqueda paralela [4, 8].

4.2 Chips especializados

Para alcanzar velocidades de exploración de árbol imposibles en un ordenador convencional, *Deep Blue* emplea 480 chips VLSI especializados, distribuidos en los treinta nodos SP. Cada chip integra tres módulos fundamentales: un generador de movimientos basado en lógica combinatoria (matriz 8x8), una función de evaluación en dos niveles —rápida (un ciclo de reloj) y lenta (análisis columna por columna de estructuras complejas)—, y un controlador de búsqueda alpha-beta de ventana nula con extensiones locales (jaques, peones pasados⁸, detección de repeticiones en 32 plies⁹) [4]. Este diseño en silicio permite examinar, de forma acumulada, entre 2 y 2.5 millones de posiciones por segundo por chip. [8]

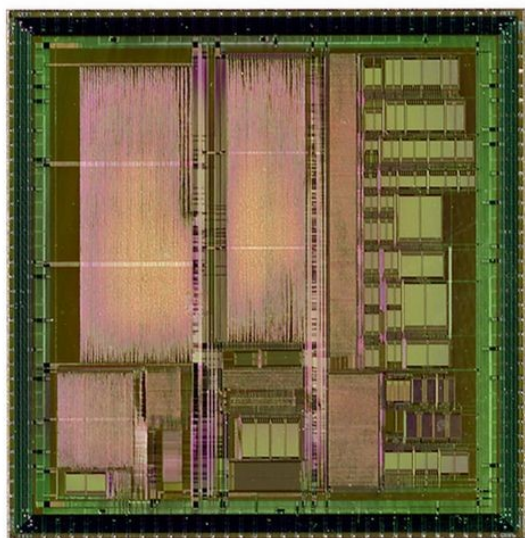


Figura 3: Esquema del chip VLSI de Deep Blue: generador de movimientos, evaluación rápida/lenta y controlador alpha-beta de ventana nula.

4.3 Jerarquía Maestro–Trabajador–Chip

La ejecución de la búsqueda se organiza jerárquicamente en tres niveles. De los procesadores SP uno es designado como maestro y el resto como trabajadores. En el nivel superior, el nodo maestro realiza las primeras iteraciones de la búsqueda, explorando las ramas superiores del árbol de juego, generando posiciones “hoja” y

⁸Un *peón pasado* es aquel que no tiene peones enemigos en su misma columna ni en las columnas adyacentes que puedan impedir su avance hacia la octava fila

⁹Un *ply* (o media jugada) corresponde al movimiento de un solo bando; dos plies equivalen a un movimiento completo de ambos jugadores

distribuyéndolas entre los nodos trabajadores. A continuación, esos nodos trabajadores profundizan el análisis unas pocas plies más, aplicando tablas de transposición y heurísticas de selección. Finalmente, cada trabajador delega las posiciones finales a un grupo de chips VLSI (16 por nodo) para que realicen en *hardware* la fase más profunda y estable de la exploración del árbol de juego. Esta arquitectura maestro-trabajador-chip maximiza el paralelismo y mantiene siempre ocupados tanto los procesadores como los chips especializados [4, 8].

5 Algoritmos de búsqueda y evaluación

La fortaleza de *Deep Blue* radica en la combinación sinérgica de un motor de búsqueda en *software* con una evaluación especializada en *hardware*. Este enfoque híbrido permite explotar al máximo tanto la flexibilidad algorítmica como la velocidad de procesamiento en silicio.

5.1 Búsqueda selectiva híbrida

El motor en C emplea el algoritmo “dual credit with delayed extensions”, que otorga *créditos* parciales a movimientos forzantes (FFP por sus siglas en inglés)¹⁰ o de elevado carácter táctico y sólo los *canjea* cuando su acumulación supera un umbral predefinido, evitando así explosiones en la exploración del árbol de juego. Para preservar la coherencia de la Variación Principal¹¹, cada bando mantiene su propia cuenta de crédito y, al extenderse uno, se descuenta el mismo importe al adversario, garantizando un “*search envelope*”¹² que impide re-búsquedas oscilantes. Gracias a este mecanismo, *Deep Blue* ajusta dinámicamente la profundidad de búsqueda en función de la complejidad táctica de la posición, llegando a profundidades efectivas de hasta cuarenta plies en las variantes más forzadas [3].

5.2 Evaluación en *hardware*

La evaluación de posiciones se realiza completamente en silicio mediante más de 8000 patrones VLSI. En un primer paso, la evaluación rápida suma en un único ciclo de reloj los valores básicos de material y posicionamiento. A continuación, la evaluación lenta analiza columna por columna estructuras posicionales complejas (como seguridad de rey, configuración de peones y control de casillas críticas) utilizando tablas parametrizables cuyos pesos fueron ajustados por herramientas automáticas y entrenamiento comparativo. Este diseño asegura que cada evaluación se ejecute en tiempo constante, sin degradar la velocidad global de búsqueda [3].

¹⁰Secuencias de movimientos en que un bando efectúa una jugada forzante (como un jaque o una captura decisiva) y el oponente está prácticamente obligado a responder de manera concreta

¹¹Secuencia de movimientos que el programa considera óptima para ambos bandos, según el análisis actual.

¹²Rango de profundidad mínima garantizada que el algoritmo debe mantener para todas las ramas relevantes, de modo que la exploración no oscile entre profundidades muy distintas en posiciones similares.

5.3 Extensiones y búsqueda quiescente

Para mitigar el “efecto horizonte”¹³ y capturar tácticas decisivas, *Deep Blue* aplica extensiones locales y búsqueda quiescente¹⁴ tanto en *hardware* como en *software*. Entre los criterios de extensión destacan la *singularidad* (cuando un movimiento es claramente superior a todas las alternativas, se extiende 1-2 plies adicionales) y los *jaques* o *peones pasados*¹⁵, que reciben tratamiento preferente para explorar líneas forzadas. La detección de repeticiones se implementa mediante un buffer circular de 32 plies en el chip, que invalida automáticamente ramas cíclicas mediante un “*fail-low*”¹⁶ [3].

La búsqueda quiescente continúa hasta que no quedan capturas críticas ni amenazas inmediatas, garantizando que sólo se evalúen posiciones *quietas* y evitando así que variantes tácticas escapadas distorsionen la valoración final.

6 Paralelismo y rendimiento

La capacidad de *Deep Blue* para explorar eficientemente espacios de estado inmensos no habría sido posible sin un diseño de paralelismo multinivel cuidadosamente articulado. A lo largo de la búsqueda, se combinan técnicas de paralelismo en *software* (que permiten dividir dinámicamente el árbol de juego entre múltiples procesadores) con un despacho asíncrono de tareas a cientos de chips especializados. Este enfoque mixto optimiza la utilización de recursos y amortigua tanto la variabilidad en los tamaños de las subramas como las latencias de comunicación, resultando en un rendimiento que, aunque superior al de cualquier sistema previo, revela asimismo los desafíos de la escalabilidad en arquitecturas de gran tamaño [4].

6.1 Paralelismo en *software*

El paralelismo en *software* de *Deep Blue* se articula en varios niveles para maximizar el aprovechamiento de los 30 nodos SP antes de delegar en *hardware*. En primer lugar, se emplea *PV-parallelism*, donde tras explorar el primer movimiento de la Variación Principal (PV) todas las alternativas pueden evaluarse en paralelo con ventanas de búsqueda desplazadas para conservar la selectividad. En segundo lugar, en nodos de *fail-high*¹⁷, se pueden lanzar en paralelo búsquedas de profundi-

¹³Fenómeno que puede ocurrir en juegos de estrategia, especialmente en el ajedrez, donde se limita la capacidad de analizar las consecuencias de una jugada a un número finito de pasos debido a la complejidad del juego.

¹⁴Algoritmo que se utiliza habitualmente para extender la búsqueda en nodos inestables en árboles de juego minimax en programas informáticos.

¹⁵Un peón pasado es aquel que no puede ser detenido por peones enemigos en su misma o en las columnas adyacentes.

¹⁶En las variantes de poda alfa-beta, *fail-low* ocurre en un nodo cuando ninguna de las jugadas exploradas logra elevar el valor evaluado por encima de la cota inferior α . Este resultado suele emplearse para terminar la exploración precozmente, ahorrando cómputo en ramas que no pueden mejorar la decisión actual.

¹⁷Un *fail-high* se produce cuando, en la poda alfa-beta, la evaluación de un nodo supera la cota superior β , indicando que ya existe una jugada conocida que anulará esa posición y, por tanto, no es necesario explorar el resto de sus sucesores.

dad reducida para refinar amenazas y extensiones, mientras que en nodos de *fail-low* todas las ramas se procesan concurrentemente hasta encontrar un corte. Finalmente, la sincronización global se realiza al término de cada iteración de búsqueda, garantizando consistencia de las tablas de transposición y de los datos de análisis antes de pasar a la siguiente profundidad [4].

6.2 Offloading a hardware

Una vez exploradas varias capas en *software*, *Deep Blue offloads*¹⁸ de forma asíncrona las posiciones finales a los 480 chips VLSI. Cada nodo SP puede gestionar múltiples búsquedas de chip en paralelo, enviando tareas y recibiendo resultados sin bloquearse en espera de la finalización de cada tarea [4]. Para mantener un balance de carga adecuado, las búsquedas en *hardware* que exceden las 8000 posiciones expandidas pueden abortarse y reencolarse en *software*, permitiendo dividirlos en subproblemas más pequeños y redistribuirlos entre otros nodos SP [4].

6.3 Medición de eficiencia

Los experimentos de rendimiento con una configuración de un solo nodo SP (24 chips) mostraron eficiencias del 75 % en posiciones relativamente quietas y del 30 % en aquellas con tácticas profundas, reflejando la capacidad de paralelizar ramas homogéneas y de adaptarse a nodos muy desequilibrados [4]. Al escalar al sistema completo de 30 nodos, se estimó una eficiencia global de aproximadamente 12 % en posiciones tranquilas y 8 % en posiciones tácticas, debido principalmente a la sobrecarga de comunicación y a la variabilidad en el tamaño de las subramas [4].

7 Componentes de apoyo

Los elementos auxiliares que complementan la búsqueda de *Deep Blue* (libro de aperturas, libro extendido, bases de finales y control de tiempo) fueron determinantes para integrar conocimiento posicional experto con la potencia de cálculo. Estos componentes permitieron al sistema navegar con confianza en fases de la partida donde la fuerza bruta pierde eficiencia, garantizando solidez en las aperturas, exactitud en los finales y un uso óptimo del tiempo de cómputo disponible [4].

7.1 Libro de aperturas

El libro de aperturas de *Deep Blue* fue ensamblado manualmente por el Gran Maestro Joel Benjamin, junto con los Grandes Maestros Nick De Firmian, John Fedorowicz y Miguel Illescas, y contenía aproximadamente 4 000 posiciones seleccionadas según la experiencia práctica del sistema [4]. Cada posición del libro fue comprobada mediante ejecuciones nocturnas de *Deep Blue* para asegurar que las líneas

¹⁸En este contexto, “*offload*” hace referencia a la transferencia asíncrona de tareas de búsqueda finales desde el procesador principal a los chips VLSI especializados, delegando en estos aceleradores el cómputo intensivo para aprovechar sus capacidades de evaluación en silicio.

propuestas resultaran razonables bajo el motor de búsqueda, priorizando aquellas aperturas en las que el sistema demostraba mayor robustez táctica. Antes de cada encuentro, se elegía un repertorio específico en función de la situación del *match* y de la elección de color, incluyéndose un pequeño “*override book*”¹⁹ de última hora para correcciones de último minuto [4].

7.2 *Extended Book*

Cuando una posición no estaba cubierta por el libro principal, *Deep Blue* empleaba un *extended book* basado en un repositorio de cerca de 700 000 partidas de Grandes Maestros [4]. A cada jugada candidateada se le asignaba una bonificación escalar calculada a partir de la frecuencia de aparición en la base de datos, la fuerza de los jugadores que la emplearon, los resultados estadísticos y las anotaciones, de manera no lineal y ponderada para reflejar consenso teórico de apertura [4]. En situaciones extremas, una bonificación superior a medio peón²⁰ permitía ejecutar la jugada directamente sin activación de la búsqueda completa [4].

7.3 Bases de datos de finales

Para los finales, *Deep Blue* recurría a bases de datos de finales con todas las posiciones de hasta cinco piezas y casos seleccionados de seis piezas (por ejemplo, peones bloqueados), extraídas principalmente de Colecciones de Ken Thompson [16] y contribuciones de Lewis Stiller [4]. Cada posición se indexaba como un bit (ganar o no perder), de modo que al alcanzarse durante la búsqueda se aplicaba inmediatamente un *cutoff* sin necesidad de desarrollar subárboles adicionales según el valor del resultado, compuesto por dos partes: *high-order* para la victoria o empate, *low-order* para el desempate. [4]. Estas bases residían localmente en cada nodo SP y en arreglos RAID²¹ compartidos, garantizando acceso rápido en tiempo de búsqueda [4].

7.4 Control de tiempo

El mecanismo de control de tiempo de *Deep Blue* definía dos objetivos temporales antes de cada movida: un tiempo “normal”, calculado como el tiempo restante hasta el próximo control dividido por los movimientos estimados, y un tiempo “pánico” equivalente a un tercio del tiempo restante [4]. Si al alcanzar el tiempo normal no se detectaban condiciones excepcionales, el sistema jugaba el mejor movimiento

¹⁹Un *override book* es un subconjunto de aperturas y líneas de juego añadido justo antes de la partida para corregir o anular jugadas del libro principal, incorporando ajustes de última hora basados en preparaciones específicas contra el adversario.

²⁰En ajedrez computacional, “medio peón” (*half pawn*) equivale a 50 centipawns; un centipawn es la centésima parte del valor de un peón estándar y sirve para medir diferencias de valoración posicional con mayor granularidad.

²¹Un grupo/matriz redundante de discos independientes (también, RAID, del inglés *redundant array of independent disks*) hace referencia a un sistema de almacenamiento de datos que utiliza múltiples unidades, entre las cuales se distribuyen o replican los datos.

disponible; en caso de variación excesiva en el valor del movimiento (más de 15 centipawns) o estados de *fail-low/high*, la búsqueda continuaba hasta nueva resolución o hasta el tiempo pánico, circunstancia que solo ocurrió una vez en todo el match de 1997 [4].

8 Impacto y legado

La victoria de *Deep Blue* en 1997 marcó un antes y un después en la historia de la inteligencia artificial y la supercomputación. Por primera vez, una máquina derrotaba bajo control reglamentario a un campeón mundial de ajedrez, demostrando de modo fehaciente que arquitecturas masivamente paralelas y *hardware* especializado podían resolver problemas de búsqueda combinatoria de gran escala [10, 6]. Este suceso desencadenó un debate sobre los límites de la creatividad mecánica y la naturaleza de la inteligencia, al tiempo que consolidó la confianza institucional y mediática en la investigación aplicada en IA.



Figura 4: El campeón mundial de ajedrez Garry Kasparov (izq.) realiza un movimiento durante su cuarta partida contra *Deep Blue*.

El hecho generó titulares en todo el mundo y revalorizó la supercomputación como herramienta de investigación científica. Medios como *Scientific American* analizaron las implicaciones del triunfo, señalando cómo el empleo de 480 chips VLSI y 30 nodos RS/6000 SP redefinió las expectativas sobre el rendimiento de las máquinas [6]. La repercusión académica fue inmediata: Carnegie Mellon otorgó el Premio Fredkin, valorado en 100 000 USD, y se multiplicaron los congresos y talleres dedicados a la paralelización extrema y al diseño de aceleradores especializados [10].

8.1 Influencias posteriores

Tras el éxito de *Deep Blue*, el paradigma de desplazar cargas críticas de cálculo a hardware dedicado se expandió rápidamente. En el ámbito financiero, los *hedge funds* adoptaron técnicas de *cluster computing* para simulaciones de riesgo en tiempo real, inspiradas en la arquitectura maestro-trabajador-chip de IBM [1]. Asimismo,

la práctica de *offloading* a aceleradores (ahora GPUs y FPGAs) se consolidó en aplicaciones de aprendizaje profundo, siguiendo el modelo pionero de *Deep Blue* [5].

8.2 Repercusiones histórico-culturales

El enfrentamiento humano-máquina suscitó reflexiones en filosofía y sociología de la ciencia sobre la confianza en sistemas automatizados. Documentales como *Game Over: Kasparov and the Machine* (2003) examinaron las acusaciones de falta de transparencia en el diseño de *Deep Blue*, alimentando la demanda de auditorías de “caja blanca” en sistemas críticos. A largo plazo, esta preocupación influyó en la creación de estándares de explicabilidad en IA, que hoy exigen trazabilidad de decisiones algorítmicas en sectores como la sanidad y la justicia [11, 2].

8.3 Curiosidades

Durante la cuarta partida de 1997, *Deep Blue* ejecutó un movimiento inesperado (que Kasparov calificó de “creatividad” ante lo insólito), el cual años después se atribuyó a un desajuste en los parámetros de evaluación, no a un cálculo “creativo” de la máquina [12]. Este incidente ilustró la delgada línea entre la aparente originalidad de una IA y la complejidad de sus reglas internas, subrayando la importancia de la transparencia para comprender sus decisiones.

9 Conclusiones

La experiencia de *Deep Blue* pone de manifiesto cómo la confluencia de tres líneas de innovación (*hardware* VLSI especializado, algoritmos de búsqueda selectiva y bases de datos inteligentes) permitió superar las barreras históricas entre el ajedrez humano y la computadora. En primer lugar, los chips VLSI diseñados *ad hoc* para generación de movimientos y evaluación en un ciclo de reloj demostraron que el procesamiento masivamente paralelo podía explotar la fuerza bruta sin sacrificar precisión táctica [4]. En segundo lugar, el algoritmo de “*dual credit with delayed extensions*” introdujo una búsqueda altamente no uniforme que adaptaba dinámicamente la profundidad según la complejidad de la posición, con extensiones fraccionarias que prevenían explosiones combinatorias y preservaban robustez frente a amenazas tácticas [1]. Por último, la integración de un libro principal, un “*extended book*” de 700 000 partidas y bases de finales con todos los casos de hasta cinco piezas proporcionó un conocimiento posicional sólido que complementó el poder de cómputo, reduciendo la dependencia en el cálculo exhaustivo en fases teóricamente bien entendidas [6].

Históricamente, la victoria de *Deep Blue* sobre Garry Kasparov en 1997 simboliza el tránsito de la IA simbólica y de fuerza bruta hacia sistemas híbridos que combinan aprendizaje experto con arquitecturas de alto rendimiento. Este hito no sólo impulsó la investigación en supercomputación y en aceleradores especializados (preludiando el uso extensivo de GPUs y FPGAs en IA), sino que también abrió el camino para metodologías que hoy sustentan desde la biología computacional hasta el análisis financiero en tiempo real [13]. En suma, *Deep Blue* demostró que el equilibrio entre conocimiento experto, paralelismo extremo y algoritmos selectivos puede superar la fuerza bruta pura, marcando el umbral de la era moderna de la inteligencia artificial aplicada.

Referencias

- [1] Lawrence Aung. Deep blue: The history and engineering behind computer chess. USC Viterbi School of Engineering, <https://illumin.usc.edu/deep-blue-the-history-and-engineering-behind-computer-chess/>, March 2010. Updated November 12, 2017; Accessed: 2025-05-09.
- [2] Paolo Bory. Deep new: The shifting narratives of artificial intelligence from deep blue to alphago. *Convergence*, 25(4):627–642, 2019.
- [3] M Campbell, A Joseph Hoane, and Feng-hsiung Hsu. Search control methods in deep blue. In *AAAI Spring Symposium on Search Techniques for Problem Solving under Uncertainty and Incomplete Information*, pages 19–23, 1999.
- [4] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.
- [5] Forbes Technology Council. What deep blue and alphago can teach us about explainable ai. *Forbes*, May 2019. Accessed: 2025-05-09.
- [6] Larry Greenemeier. 20 years after deep blue: How ai has advanced since conquering chess. *Scientific American*, June 2017. Accessed: 2025-05-09.
- [7] F-h Hsu, Thomas S Anantharaman, Murray S Campbell, and Andreas Nowatzky. Deep thought. In *Computers, Chess, and Cognition*, pages 55–78. Springer, 1990.
- [8] Feng-hsiung Hsu. Ibm’s deep blue chess grandmaster chips. *IEEE micro*, 19(2):70–81, 1999.
- [9] Feng-hsiung Hsu, Thomas Anantharaman, Murray Campbell, and Andreas Nowatzky. A grandmaster chess machine. *Scientific American*, 263(4):44–51, 1990.
- [10] IBM. Deep blue. <https://www.ibm.com/history/deep-blue>, 2025. Accessed: 2025-05-09.
- [11] Richard E Korf. Does deep blue use ai? In *Deep Blue Versus Kasparov: The Significance for Artificial Intelligence*, pages 1–2, 1997.
- [12] Jennifer Latson. Did deep blue beat kasparov because of a system glitch? *TIME*, February 2015. Accessed: 2025-05-09.
- [13] Steven Levy. What deep blue tells us about ai in 2017. *WIRED*, May 2017. Accessed: 2025-05-09.
- [14] Monty Newborn and Danny Kopec. Results of acm’s eighteenth computer chess championship. *Communications of the ACM*, 31(8):992–995, 1988.
- [15] Claude E Shannon. Xxii. programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(314):256–275, 1950.

- [16] Ken Thompson. Retrograde analysis of certain endgames. *ICGA Journal*, 9(3):131–139, 1986.
- [17] Alan M Turing. Digital computers applied to games. *Faster than thought*, 1953.