

Universidad Internacional de La Rioja
Facultad de Ciencias de la Educación y Humanidades
Máster Universitario en Formación del Profesorado de
Educación Secundaria Obligatoria y Bachillerato, Formación
Profesional y Enseñanzas de Idiomas

**Diseño de situación de aprendizaje a
través del ajedrez para la asignatura
Programación. Ciclo de Formación Grado
Superior en desarrollo de aplicaciones web
(CFGS-DAW).**

Trabajo fin de estudio presentado por:	Ángel Jesús Hita Barraza
Tipo de trabajo:	Programación didáctica
Especialidad:	Tecnología
Director/a:	Estefanía Sánchez Castellano
Fecha:	05/02/2026

Agradecimientos

Quería dar las gracias a Leontxo García, por su divulgación que inspiró este trabajo.

A Gema Hita, ojalá ser tan buena maestra como ella.

A Ignacio Barraza, ojalá ser tan buen ajedrecista como él.

Resumen

El uso de la IA en nuestra sociedad no para de crecer y los estudiantes deberían, en algún momento de su enseñanza, entender qué ocurre dentro de una IA y qué mecanismos utilizamos para crearla. Bajo esta idea, este trabajo es una propuesta de una situación de aprendizaje para el desarrollo práctico de una inteligencia artificial capaz de jugar al ajedrez utilizando el lenguaje JavaScript. Se ha utilizado el ajedrez como herramienta transversal en el proyecto por sus ventajas pedagógicas, como recogen diversos estudios. Para la implementación de la situación de aprendizaje, se utilizan metodologías activas, como el aprendizaje basado en proyectos y la gamificación mediante la competición. Esta combinación permite que el alumnado asimile fundamentos algorítmicos complejos, como los árboles de búsqueda y las estructuras de datos, potenciando su pensamiento computacional. Por último, también se tienen en cuenta los principios del diseño universal para el aprendizaje, garantizando que el reto sea escalable y atienda a la diversidad, adaptándose a las necesidades del alumnado.

Palabras clave: (Máximo 5 palabras)

Ajedrez, inteligencia artificial, gamificación, DUA, ABP.

Abstract

The use of AI in our society continues to grow, and students should, at some point in their education, understand what happens inside an AI and what mechanisms we use to create it. With this in mind, this work proposes a learning situation for the practical development of an artificial intelligence capable of playing chess using the JavaScript language. Chess has been used as a cross-curricular tool in the project due to its pedagogical advantages, as documented in various studies. Active methodologies, such as project-based learning and gamification through competition, are used to implement the learning situation. This combination allows students to assimilate complex algorithmic fundamentals, such as search trees and data structures, enhancing their computational thinking. Finally, the principles of Universal Design for Learning are also taken into account, ensuring that the challenge is scalable and addresses diversity, adapting to the needs of the students.

Keywords:

Chess, artificial intelligence, gamification, UDL, PBL.

Índice de contenidos

1. Introducción	8
1.1. <i>Justificación y planteamiento del problema</i>	<i>10</i>
1.2. <i>Objetivos</i>	<i>11</i>
1.2.1. Objetivo general	11
1.2.2. Objetivos específicos	12
2. Programación didáctica	13
2.1. <i>Presentación de la Programación didáctica</i>	<i>13</i>
2.2. <i>Contextualización</i>	<i>15</i>
2.3. <i>Elementos curriculares</i>	<i>16</i>
2.3.1. Competencias profesionales	17
2.4. <i>Metodología</i>	<i>19</i>
2.5. <i>Organización de los espacios de aprendizaje</i>	<i>20</i>
2.6. <i>Distribución del tiempo</i>	<i>21</i>
2.6.1. Temporalización general del módulo profesional	21
2.6.2. Cronograma de la Unidad de Trabajo del proyecto final IA de ajedrez	22
2.7. <i>Selección y organización de los recursos y materiales</i>	<i>23</i>
2.7.1. Recursos físicos y espaciales	23
2.7.2. Recursos tecnológicos y software	23
2.7.3. Recursos didácticos elaborados por el docente	24
2.8. <i>Atención a la diversidad</i>	<i>24</i>
2.8.1. Casos específicos de atención a la diversidad	26
2.9. <i>Evaluación</i>	<i>26</i>
2.10. <i>Situación de aprendizaje/ unidad de trabajo</i>	<i>29</i>
2.10.1. Sesiones de la unidad de trabajo	29
2.11. <i>Evaluación de la Programación didáctica</i>	<i>39</i>
2.11.1. Análisis DAFO de la situación de aprendizaje	39
2.11.2. Instrumentos de Autoevaluación Docente	41
3. Conclusiones	43
4. Limitaciones y prospectiva	45

Anexos.....	49
-------------	----

Índice de figuras

Figura 1.	Fórmula del resultado esperado en una partida	14
Figura 2.	Fórmula del resultado del rating Elo al finalizar la partida	14
Figura 3.	Gráfico análisis DAFO.....	41
Figura 4.	Ejemplo diana de autoevaluación.....	42

Índice de tablas

Tabla 1.	Referencias del curriculum	18
Tabla 2.	Temporalización anual del módulo	21
Tabla 3.	Cronograma de las sesiones	22
Tabla 4.	Criterios de evaluación	27
Tabla 5.	Rúbrica de evaluación	27
Tabla 6.	Rúbrica de autoevaluación diagnóstica para el alumnado.....	28
Tabla 7.	Sesión 1 del proyecto	29
Tabla 8.	Sesión 2 del proyecto	30
Tabla 9.	Sesión 3 del proyecto.	31
Tabla 10.	Sesión 4 del proyecto	32
Tabla 11.	Sesión 5 del proyecto	33
Tabla 12.	Sesión 6 del proyecto	34
Tabla 13.	Sesión 7 del proyecto	35

Tabla 14.	Sesión 8 del proyecto	36
Tabla 15.	Sesión 9 del proyecto	36
Tabla 16.	Evaluación de los resultados de aprendizaje de la unidad de trabajo.	37
Tabla 17.	Elaboración de autoevaluación docente	42

1. Introducción

A mediados de la década de los 60, el pedagogo de origen húngaro László Polgár, escribía cartas con la que sería su futura mujer Klára, acerca de un proyecto pedagógico sobre el talento. En este proyecto, László sostenía que el talento no era innato, sino que se podía educar y enseñar. Durante esas conversaciones uno de los debates que tuvo la pareja Polgár fue qué disciplina elegir para probar el experimento, se barajaron varias, como artes, idiomas o ajedrez. La opción elegida fue ajedrez porque necesitaban una disciplina en la que el éxito fuera 100% objetivo y evaluable. El resultado es de sobra conocido, sus hijas, las hermanas Polgár (Judit, Zsófia y Zsuzan) tuvieron un éxito nunca antes visto en el mundo del ajedrez, siendo Zsuzan campeona del mundo de ajedrez en 1996 y Judit la jugadora con más Elo fide (herramienta que se usa para evaluar la fuerza de un jugador de ajedrez) de la historia de ajedrez. Del caso de las hermanas Polgár, László Polgár (1989) escribiría *Cómo criar a un genio*. Hoy en día hay innumerables artículos bibliográficos probando el éxito del ajedrez como herramienta educativa y pedagógica.

Inspirado en esta historia y utilizando el ajedrez como herramienta transversal, este trabajo final de estudios tiene como objeto el diseño de una situación de aprendizaje, enmarcada en el módulo de programación del ciclo formativo de grado superior en desarrollo de aplicaciones web. La propuesta se fundamenta en la metodología de aprendizaje basado en proyectos, que plantea a los estudiantes el reto de desarrollar una inteligencia artificial que sea capaz de jugar al ajedrez utilizando el lenguaje JavaScript.

El plantearlo a través del desarrollo de una inteligencia artificial no es arbitrario, en el contexto educativo actual, la alfabetización en IA se ha convertido en una necesidad en nuestra sociedad. Informes de la UNESCO (2023) destacan la urgencia para que el alumnado comprenda no sólo cómo usar la IA, sino cómo funciona, sus algoritmos subyacentes y sus implicaciones éticas. Este proyecto responde a esa necesidad, alineándose con las competencias de "*comprensión, uso y desarrollo de IA*" sugeridas por la UNESCO, pero aplicadas al nivel de la formación profesional. La creación práctica de un bot de ajedrez permite trascender la visión de la IA como una máquina que da respuestas, sin cuestionar el procedimiento para encontrarlas, a una comprensión profunda mediante la metodología del construccionismo, la cual postula que el

aprendizaje es más eficaz cuando el estudiante construye un objeto tangible y significativo. Al enfrentarse al reto de programar una IA, el alumnado transita por el modelo progresivo de aprendizaje de *"Usar-Modificar-Crear"*. No se limitan a interactuar con herramientas existentes, sino que deben deconstruir problemas complejos mediante el pensamiento computacional, aplicando conceptos de abstracción, descomposición y diseño de algoritmos para que la máquina pueda tomar decisiones.

Desde una perspectiva técnica, el desarrollo de este proyecto obliga al alumnado a asimilar los fundamentos de la IA, es decir, el razonamiento basado en reglas, utilizando condiciones y árboles de decisión para evaluar jugadas. Al programar la lógica que permite a la IA evaluar posiciones y seleccionar el mejor movimiento, los alumnos comprenden de primera mano cómo los algoritmos procesan la información y cómo la calidad de las reglas o los datos introducidos determinan la inteligencia de la máquina. Este enfoque práctico es fundamental para entender lo que hay dentro de una IA: al ver cómo sus propias líneas de código generan una "inteligencia" capaz de competir, los estudiantes adquieren una visión crítica sobre las limitaciones, la posibilidad de sesgos algorítmicos y la responsabilidad del desarrollador en la respuesta final del algoritmo. De este modo, la IA deja de ser un concepto abstracto para convertirse en una competencia profesional tangible, integrando saberes matemáticos y lógicos esenciales para el perfil del grado que están cursando.

Desde una perspectiva didáctica, utilizar el ajedrez como herramienta transversal ayuda a potenciar los beneficios que aporta al alumnado. Además, puede servir como vínculo de unión entre todas las asignaturas, lo cual refuerza la mejor comprensión del aprendizaje. Cognitivamente, entrena el pensamiento analítico y la resolución de problemas, habilidades transferibles a la lógica computacional y las matemáticas según autores como Bart (2014). Esto facilita la comprensión de algoritmos complejos y estructuras de datos, fundamentales en el perfil de DAW. Por otro lado, fomenta la disciplina, la concentración y la toma de decisiones bajo presión, ayudando a los estudiantes a entender las consecuencias lógicas de sus acciones, tal como destaca Susan Polgár. Su uso como eje vertebrador permite atender a la diversidad, ofreciendo retos escalables que mantienen la motivación intrínseca y el rigor académico en todas las asignaturas.

El desarrollo de esta programación didáctica se rige por la normativa vigente, específicamente la LOMLOE ([Ley Orgánica 3/2020](#)) define la ley orgánica de educación y para el estado español se especifica en el [Real Decreto 500/2024](#), de 21 de mayo, establece los títulos de Formación Profesional de grado superior y se fijan sus enseñanzas mínimas y el [Real Decreto 405/2023](#), que actualiza el título de Técnico Superior en DAW. Para la situación que abordaremos el diseño de esta unidad de aprendizaje, nos regiremos por la normativa autonómica del País Vasco, en concreto el [decreto 245/2011](#), de 29 de noviembre, por el que se establece el currículo correspondiente al título de técnico superior en desarrollo de aplicaciones web. Por último, cabe destacar que la metodología ABP permitirá atender a la diversidad del aula, ofreciendo un reto escalable donde cada alumno puede profundizar en la complejidad del algoritmo según su ritmo de aprendizaje, fomentando así la autonomía, la creatividad y la resolución de problemas.

1.1. Justificación y planteamiento del problema

La rápida evolución del sector tecnológico plantea un desafío constante para la Formación Profesional, especialmente en una titulación plenamente digital como es el grado superior en desarrollo de aplicaciones web. Como hemos comentado anteriormente, informes recientes de la UNESCO (2023), exponen que existe una urgencia global por fomentar la alfabetización digital en el uso de la inteligencia artificial, la cual no debe limitarse al uso de herramientas, sino profundizar en la comprensión de los algoritmos, la gestión de datos y las implicaciones éticas.

El problema didáctico para abordar este trabajo es ejemplificar la enseñanza de conceptos abstractos de programación —como la recursividad, las estructuras de datos complejas (árboles, grafos) y la optimización algorítmica— de una manera más intuitiva y práctica utilizando el ajedrez y profundizar al mismo tiempo en el concepto de lo que hay detrás de una inteligencia artificial. La enseñanza tradicional a menudo falla en conectar estos conceptos técnicos con aplicaciones prácticas motivadoras. Para solucionar esto, se propone el reto de la creación de una inteligencia artificial capaz de jugar al ajedrez.

La elección del ajedrez como herramienta transversal se justifica por su histórica relación con las ciencias de la computación y su capacidad para modelar procesos de toma de decisiones. Autores como Bart (2014) sostienen que el ajedrez promueve habilidades de alto

nivel esenciales para un programador, tales como el reconocimiento de patrones, la evaluación de posiciones y el pensamiento crítico.

Desde una perspectiva pedagógica, el ajedrez es una herramienta que entrena habilidades vitales para el desarrollo profesional, como el análisis profundo antes de la toma de decisiones y la comprensión de las consecuencias lógicas de cada acción (S. Polgar, 2018), competencias transversales clave en el desarrollo de software.

Finalmente, esta propuesta reivindica el papel del docente como programador y diseñador de experiencias de aprendizaje, tal y como exige la normativa vigente. Desde esta perspectiva el uso de nuevas tecnologías para la creación de una IA es un enfoque innovador que potencia la motivación del alumnado y genera cambios profundos y duraderos que resuelven necesidades reales del contexto educativo, mejorando significativamente la calidad de la enseñanza. No ha sido sencillo encontrar referencias sobre la alfabetización digital en inteligencia artificial, debido a lo novedoso de la herramienta, todavía no hay escrita mucha literatura sobre el tema, no así, en cambio, sobre el aprendizaje basado en proyectos. Un trabajo muy destacable es el de Poot-Delgado (2013), titulado "*Retos del aprendizaje basado en problemas*". El autor explica la mecánica del ABP de forma clara: El ABP es un problema real que sirve para detectar qué es exactamente lo que los alumnos necesitan aprender. A partir de ahí, se busca y analiza la información para luego volver al escenario inicial y resolverlo. En la práctica, es un esquema excelente para darle sentido a lo que se estudia y fomentar la resolución real de conflictos.

Este modelo también tiene sus contrapartidas, y el estudio no las ignora. Dado que el ABP obliga a los estudiantes a tomar la responsabilidad de investigar, trabajar en equipo y enfrentarse a situaciones del mundo real, las dificultades individuales son inevitables. Lo más valioso de esta investigación es que describe estos obstáculos en un contexto muy específico y ofrece consejos muy prácticos para que el profesorado pueda afrontarlo y superarlo sin frustrarse en el intento.

1.2. Objetivos

1.2.1. Objetivo general

Diseñar una situación de aprendizaje innovadora fundamentada en el aprendizaje basado en proyectos, en el contexto de una programación didáctica anual para la asignatura de

programación del ciclo formativo de grado superior en desarrollo de aplicaciones web, cuyo eje vertebrador sea el desarrollo de una inteligencia artificial de ajedrez en lenguaje JavaScript, con el fin de facilitar la asimilación de fundamentos algorítmicos complejos y potenciar el pensamiento computacional del alumnado.

1.2.2. Objetivos específicos

Para alcanzar el objetivo general de diseñar una situación de aprendizaje basada en la creación de una IA de ajedrez, se establecen los siguientes objetivos específicos:

1. Diseñar actividades basadas en el aprendizaje basado en proyectos con el uso de ajedrez que permitan desarrollar las competencias laborales del CFGS-DAW.
2. Crear instrumentos de evaluación que permitan utilizar el ajedrez en una situación de aprendizaje.
3. Investigar y analizar conceptos de IA para una situación de aprendizaje.

2. Programación didáctica

2.1. Presentación de la Programación didáctica

Los alumnos deberán desarrollar una IA de ajedrez a través de un proceso de investigación, planificación, diseño y programación. Al plantear a los alumnos el reto de crear una IA capaz de jugar al ajedrez, les obligará a investigar y aplicar conocimientos de programación para resolver el problema. El modo de testear la IA de ajedrez será haciéndola competir entre otras IAs y las IAs de los compañeros, esto les evaluará con un rating Elo que les servirá para su evaluación final. Los alumnos tendrán un rol activo y responsable de su aprendizaje, teniendo que tomar decisiones sobre los algoritmos y el diseño de la IA, y debiendo dedicar tiempo a investigar y buscar información de cómo funciona la lógica en ajedrez para elegir la mejor jugada. También tendrán el rol de colaboración a la hora de tener que evaluar la IA compitiendo con la de sus compañeros y mejorar su propio código en función del resultado de las partidas. El profesor deberá orientar a los alumnos en el proceso, resolviendo dudas y proporcionando recursos. Además, el profesorado de otras asignaturas tendrá el papel de motivar de cara a este proyecto cuando utilicen el ajedrez como herramienta transversal. Por último, durante el desarrollo del proyecto el docente tendrá la función de evaluador para guiar en el desarrollo correcto de la IA. Aunque el desarrollo de la IA sea a título individual, los alumnos tendrán que formar grupos para probar la IA con la de sus compañeros y, después de que compitan entre ellas, darse feedback para mejorar su código. En cuanto a los recursos se utilizará el aula de informática para escribir el código, tableros físicos para evaluar el razonamiento de las jugadas. Como recursos digitales se usará Visual Studio Code para el desarrollo de código, nodeJS como lenguaje de programación (JavaScript) y GitHub para gestión y publicación del código.

La integración de la gamificación en el aprendizaje basado en proyectos y la creación de una inteligencia artificial de ajedrez aporta beneficios a los alumnos ya que transforma el aprendizaje de la programación en una experiencia constructivista, dinámica y atractiva. Según Villacís Macías et al. (2022), el uso de mecánicas lúdicas (como la competición entre inteligencias artificiales) estimula la motivación intrínseca del estudiante, fomentando su autonomía, imaginación, creatividad y su capacidad para investigar y resolver problemas de forma independiente. Por otro lado, esta estrategia permite dejar atrás la evaluación

tradicional para dar paso a un entorno de competencia sana y trabajo en equipo entre estudiantes que promueve la exploración, reduce el aburrimiento y facilita la asimilación de conocimientos, lo que se ve reflejado en la mejora del rendimiento académico.

Se ha explicado que se utilizará el sistema de rating Elo como métrica de evolución y evaluación de los estudiantes, pero no se ha aclarado qué es este sistema. El sistema Elo es un método de clasificación empleado en el mundo del ajedrez (entre otros deportes) para medir el nivel de habilidad y el rendimiento de los jugadores basándose en los resultados de sus competiciones. El sistema es un método matemático (creado por Arpad Elo) utilizado para medir la habilidad relativa de los jugadores en juegos competitivos de suma cero. En cada partida los jugadores ganan o pierden puntos de Elo en función del resultado de la partida, este intercambio de puntos se utilizando una fórmula del resultado esperado de la partida:

Figura 1. *Fórmula del resultado esperado en una partida*

$$E_A = \frac{1}{1 + 10^{\frac{(R_B - R_A)}{400}}}$$

Fuente: Wikipedia

Siendo cada valor:

- E_A : Puntuación esperada del Jugador A (un valor entre 0 y 1, que representa su probabilidad de ganar).
- R_A : Elo actual del jugador A.
- R_B : Elo actual del jugador B.

Se divide entre 400 porque es el valor donde un jugador con 400 puntos más de Elo que su rival tendría más de 90% de probabilidad de ganar.

Una vez termina la partida se comparan el resultado real con el resultado esperado para saber cuántos puntos se suman o se restan:

Figura 2. *Fórmula del resultado del rating Elo al finalizar la partida*

$$R'_A = R_A + K \cdot (S_A - E_A)$$

Fuente: Wikipedia

Siendo cada valor:

- R'_A : El nuevo Elo del Jugador A tras la partida.

- R_A : El Elo anterior del Jugador A.
- K : El "Factor K" (o coeficiente de desarrollo). Es la cantidad máxima de puntos que se pueden ganar o perder en una partida. En el ajedrez, suele ser de 40 para jugadores nuevos, 20 para jugadores habituales y 10 para los grandes maestros (para que su puntuación sea más estable).
- S_A : El resultado real de la partida. Vale 1 si el Jugador A gana, 0.5 si hay empate (tablas) y 0 si el Jugador A pierde.
- E_A : El resultado esperado que calculamos en la primera fórmula.

La justicia de este sistema está en que evalúa el rendimiento basándose en la diferencia directa entre el resultado que obtuviste en la partida y el que estadísticamente se esperaba. De este modo, el sistema premia de forma muy generosa las victorias improbables contra rivales muy superiores y apenas penaliza las derrotas cuando ya se pronosticaba que se iba a perder. A su vez el sistema también es exigente con los jugadores fuertes, quienes pueden llegar a perder puntos incluso haciendo tablas, ya que ese resultado representa un rendimiento inferior la alta probabilidad matemática de victoria.

En lo que respecta a este proyecto, el sistema Elo es beneficioso porque actúa como una herramienta de gamificación que evalúa las inteligencias artificiales creadas por los estudiantes haciéndolas competir entre ellas. Este rating proporciona una evidencia que sirve como un instrumento de evaluación real, objetivo y justo del funcionamiento del código. Además, al calcularse el Elo en distintas fases del proyecto, no solo se evalúa el resultado final del trabajo, sino que permite medir la progresión del alumno, facilitando que cada estudiante reciba feedback durante el proceso y avance en el desarrollo del algoritmo según su propio ritmo y capacidad.

2.2. Contextualización

El diseño de esta programación se plantea en un contexto educativo realista. Será en un instituto donde se imparte formación profesional en la provincia de Bizkaia. Concretamente en la ciudad de Gernika. El instituto, de carácter público, está situado en un entorno urbano con un nivel socioeconómico medio, el cual cuenta con aulas de informática equipadas para

el desarrollo de software. El grupo destinatario está formado por 25 estudiantes de primer curso de DAW. Se trata de un grupo heterogéneo en cuanto a edad, mayoritariamente adolescentes (18-22 años), y en cuanto a procedencia académica (bachillerato tecnológico y grado medio de sistemas microinformáticos y redes).

Se contempla la atención a la diversidad como un eje transversal. Específicamente, dentro del grupo clase se identifica un posible caso de alumna con altas capacidades y con experiencia en la programación y conocimientos profundo del ajedrez, tiene el título de maestra FIDE femenina (WFM). Para esta alumna, la metodología basada en el desarrollo de la IA de ajedrez resulta especialmente útil. Alineándose con la teoría de Polgár sobre la educación de genios mediante la especialización temprana y el reto constante, el proyecto permite una escalabilidad vertical: mientras el resto de la clase trabaja en algoritmos básicos de movimiento aleatorio o captura simple, esta alumna podrá profundizar en algoritmos de búsqueda avanzada y evaluación posicional compleja, manteniendo así su motivación y evitando el aburrimiento derivado de un ritmo de aprendizaje estándar.

También se cuenta con un alumno extranjero con poca alfabetización digital que tiene dificultades para utilizar el ordenador. Este alumno necesitará una adaptación del entorno de desarrollo para que la aplicación del conocimiento teórico alcanzado en la asignatura no se vea mermado en la aplicación del proyecto.

2.3. Elementos curriculares

A nivel estatal nos regimos la LOMLOE ([Ley Orgánica 3/2020](#)), de 29 de diciembre define cómo se regula las distintas etapas educativas y para la formación profesional de grado superior, el [Real Decreto 500/2024](#), de 21 de mayo, establece los títulos de Formación Profesional de grado superior y se fijan sus enseñanzas mínimas; en concreto la asignatura se rige por el [Real Decreto 405/2023](#), de 29 de mayo, por el que se actualizan los títulos de la formación profesional del sistema educativo de técnico superior en desarrollo de aplicaciones multiplataforma y técnico superior en desarrollo de aplicaciones web.

A nivel autonómico se rige por el [decreto 245/2011](#), de 29 de noviembre, por el que se establece el currículo correspondiente al título de técnico superior en desarrollo de aplicaciones web en Euskadi.

Los objetivos del grado en desarrollo de aplicaciones web dentro de la asignatura programación (código 0485) son:

- Utilizar lenguajes, objetos y herramientas, interpretando las especificaciones para desarrollar aplicaciones web con acceso a bases de datos
- Desarrollar la creatividad y el espíritu de innovación para responder a los retos que se presentan en los procesos y organización de trabajo y de la vida personal
- Tomar decisiones de forma fundamentada analizando las variables implicadas, integrando saberes de distinto ámbito y aceptando los riesgos y la posibilidad de equivocación en las mismas, para afrontar y resolver distintas situaciones, problemas o contingencias.

Esto se enmarcan en los resultados de aprendizaje (RA 3, RA 4, RA 5 y RA 7). Como objetivos didácticos específicos se quiere trabajar: Implementar algoritmos de búsqueda y evaluación, manipular estructuras de datos (arrays, objetos, sets...), estructuración de código, técnicas de *clean code*, programación modular, reusabilidad, algoritmos de evaluación de complejidad de código (*O- notation*) y técnicas de optimización

2.3.1. Competencias profesionales

Algunas de las competencias profesionales que se trabajan con el desarrollo de la IA de ajedrez son:

- **Sección f:** Integrar contenidos en la lógica de una aplicación web, desarrollando componentes de acceso a datos adecuados a las especificaciones. Los alumnos tendrán que programar distintas funciones, con sus respectivas clases y componentes para poder desarrollar la IA
- **Sección m:** Completar planes de pruebas verificando el funcionamiento de los componentes software, desarrollados, según las especificaciones. Se verificará la calidad de la IA haciéndola competir contra otras IAs y con los resultados los alumnos podrán hacer cambios para mejorar su código
- **Sección p:** Adaptarse a las nuevas situaciones laborales, manteniendo actualizados los conocimientos científicos, técnicos y tecnológicos relativos a su entorno profesional, gestionando su formación y los recursos existentes en el aprendizaje a lo largo de la vida y utilizando las tecnologías de la información y la comunicación. El proyecto está

orientado a la creación de una IA, un modelo que está presente en la sociedad y les ayuda a entender cómo funciona esta tecnología y a poder implementarla.

- **Sección q:** Resolver situaciones, problemas o contingencias, con iniciativa y autonomía en el ámbito de su competencia, con creatividad, innovación y espíritu de mejora en el trabajo personal y en el de los miembros del equipo. Al no existir una solución final de la IA de ajedrez los alumnos solo podrán desarrollarla con creatividad e innovación.

A modo de resumen en la siguiente tabla se describen los elementos curriculares de la situación de aprendizaje:

Tabla 1. *Referencias del curriculum*

Elementos del curriculum			
Contenido	Resultados de aprendizaje	Criterios de evaluación	Competencias profesionales
Identificación de los elementos de un programa informático	• Reconoce la estructura de un programa informático, identificando y relacionando los elementos	• Se ha modificado el código de un programa para crear y utilizar variables • Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas	Integrar contenidos en la lógica de una aplicación web, desarrollando componentes de acceso a datos adecuados a las especificaciones.
Utilización de objetos.	• Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos. • Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje	• Se han escrito programas simples. • Se han incorporado y utilizado librerías de objetos. • Se han utilizado estructuras de repetición • Se han probado y depurado los programas	Desarrollar servicios para integrar sus funciones en otras aplicaciones web, asegurando su funcionalidad.
Uso de estructuras de control, desarrollo de clases	• Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos. • Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases	• Se han reconocido la sintaxis, estructura y componentes típicos de una clase. • Se han creado y utilizado conjuntos y librerías de clases. • Se han aplicado formatos en la visualización de la información.	Completar planes de pruebas verificando el funcionamiento de los componentes software, desarrollados, según las especificaciones. Integrar servicios y contenidos distribuidos en aplicaciones web, asegurando su funcionalidad.
Lectura y escritura de información Aplicación de las estructuras de almacenamiento	• Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos. • Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.	• Se han utilizado iteradores para recorrer los elementos de las listas. • Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto. • Se han realizado programas que implementen y utilicen jerarquías de clases.	Resolver situaciones, problemas o contingencias, con iniciativa y autonomía en el ámbito de su competencia, con creatividad, innovación y espíritu de mejora en el trabajo personal y en el de los miembros del equipo.

Fuente: elaboración propia.

2.4. Metodología

La presente Situación de Aprendizaje se fundamenta en un enfoque pedagógico activo y constructivista, donde el estudiante deja de ser un receptor pasivo de información para tomar agencia de su propio conocimiento. Para este propósito, se implementa una combinación de metodologías innovadoras: el aprendizaje basado en proyectos y problemas, la gamificación, y una adaptación del modelo de aula invertida (*Flipped Classroom*). A continuación, se detalla la base teórica y la aplicación práctica de cada una de ellas:

El aprendizaje basado en proyectos es el eje transversal de la actividad. Según Poot-Delgado (2013), esta metodología invierte el proceso de aprendizaje convencional: primero se presenta el problema, se identifican las necesidades de aprendizaje, se investiga la información y, finalmente, se regresa al problema para resolverlo. Bajo esta premisa, el alumnado toma la responsabilidad de investigar y trabajar frente a situaciones reales. Para el caso de este proyecto, el reto consiste en diseñar y programar una IA de ajedrez, un problema complejo que carece de una única solución algorítmica. Esto obliga a los estudiantes a investigar estructuras de datos, evaluar con creatividad cuál sería la mejor jugada y refactorizar su código de manera autónoma, logrando un aprendizaje significativo al enfrentarse a los mismos obstáculos que un desarrollador de software en el mundo laboral.

Por otro lado, la gamificación complementa el aprendizaje basado en proyecto y consigue mitigar la posible frustración frente a un reto técnico complejo. Para autores como Villacís Macías et al. (2022), la introducción de mecánicas lúdicas en el aula no solo genera experiencias de aprendizaje atractivas y placenteras, sino que estimula profundamente la motivación intrínseca, el compromiso y el trabajo en equipo de los estudiantes. En esta unidad didáctica, la gamificación se implementa haciendo competir a las diferentes inteligencias artificiales programadas por los alumnos en torneos, evaluando su rendimiento mediante el sistema de rating Elo. Esta competición hace que la evaluación sea un proceso dinámico, donde el error pierde su connotación punitiva y se convierte en feedback para que el estudiante mejore su algoritmo.

El ajedrez como herramienta de transferencia cognitiva La elección del ajedrez para esta práctica de programación se fundamenta en su alto valor pedagógico para el desarrollo del pensamiento computacional. Según el investigador William M. Bart (2014), el ajedrez exige

una secuencia cognitiva rápida de comprensión de posiciones, inducción de patrones, y formulación y evaluación de jugadas. Bart sostiene que este conjunto coordinado de habilidades cognitivas requeridas en el ajedrez se transfiere de manera directa al aprendizaje de las matemáticas y campos relacionados que exigen inducción, análisis y evaluación de fenómenos complejos. Por tanto, abstraer la lógica del ajedrez para programar una IA refuerza de forma natural la lógica algorítmica del alumnado.

Por último, hay que explicar que la metodología del aula invertida será implementada con una pequeña modificación acorde a la necesidad del proyecto. Su formato tradicional completo, que consiste en visualizar lecciones grabadas en casa para luego hacer los deberes en clase, se ha cambiado para extraer su punto más fuerte: la optimización del tiempo presencial. En el contexto de este proyecto, en clase el alumnado, con ayuda del profesor, revisará el código y estudiarán maneras de mejorarlo, dejando para trabajo en fuera del aula la implementación y revisión. De este modo, el tiempo en el aula no se consume en clases magistrales expositivas, sino que se reserva como un espacio práctico. En clase, el docente actuará como facilitador y los estudiantes se dedicarán exclusivamente a programar, depurar errores complejos, compartir soluciones con sus compañeros y testear sus inteligencias artificiales en tiempo real.

2.5. Organización de los espacios de aprendizaje

La asignatura de programación tiene un carácter puramente digital, con lo que requiere herramientas específicas (ordenador, conexión a internet, servidores, software...) para poder llevar a cabo su enseñanza. A parte de la parte digital, las clases deben ser impartidas en un espacio que permita al alumnado el resolver dudas mientras desarrolla código, por lo que a pesar de que la asignatura es puramente digital, hay que tener en cuenta que se debe impartir en un aula que cumpla con los requisitos para que los alumnos puedan estar en clase y desarrollar código al mismo tiempo. Es por ello por lo que, desde el punto de vista de la actividad de desarrollar un proyecto, se dividen los espacios en tres categorías:

1. Espacio físico: Aula de informática y recursos manipulativos. El espacio central para las sesiones presenciales será el aula de informática. El aula deberá contar con ordenadores individuales para cada alumno, así como un proyector para que el docente pueda compartir su código, mostrar explicaciones y, en el caso del proyecto, mostrar un tablero digital de ajedrez para poder visualizar las mejores jugadas que podría realizar la IA.

2. Espacio virtual: El espacio de trabajo digital, donde el alumno desarrollará el código del proyecto, estará conformado por el siguiente stack tecnológico: Se utilizará Visual Studio Code para el desarrollo del código y NodeJS como entorno de ejecución para JavaScript. Además, GitHub actuará como el espacio para la gestión y publicación del proyecto a modo de repositorio. Se le facilitará al alumno un repositorio inicial en GitHub con el código base y material audiovisual explicativo, creando así un entorno virtual donde el alumno puede acceder a los recursos, investigar y avanzar en el proyecto por su cuenta fuera del horario lectivo.
3. Espacio de aprendizaje: Plataformas para poder profundizar en el aprendizaje del ajedrez como ChessBase, chess.com o lichess. En el caso de esta última tiene una herramienta de competición entre bots, que se puede utilizar después de la práctica para que continúen manteniendo el código y mejoren el proyecto de manera independiente.

2.6. Distribución del tiempo

2.6.1. Temporalización general del módulo profesional

El módulo de Programación cuenta con una asignación horaria que se distribuye a lo largo de los tres trimestres del curso escolar. La planificación se ha estructurado en nueve unidades de trabajo secuenciales, organizadas desde los conceptos básicos y abstractos de programación hasta la aplicación de estos conceptos a través de algoritmos.

Esta distribución tiene en cuenta el calendario escolar y reserva las últimas semanas del tercer trimestre para la ejecución de la práctica de una IA de ajedrez. Al ubicar esta unidad al final del curso, se garantiza que el alumnado tenga ya los conocimientos para poder afrontarla y al mismo tiempo permite evaluar el aprendizaje del alumno durante el curso a través de la propia práctica.

A continuación, se presenta la tabla resumen con la temporalización anual del módulo:

Tabla 2. *Temporalización anual del módulo*

Trimestre	Título de la Unidad de Trabajo (UT)	Bloque de contenidos principales	Número de Sesiones
-----------	-------------------------------------	----------------------------------	--------------------

Primer Trimestre	UT 1. Introducción.	Introducción a la programación, algoritmos y diagramas de flujo. Instalación del IDE (<i>Visual Studio Code</i>). Variables y tipos de datos.	35h
	UT 2. Estructuras de control de código.	Estructuras de control (condicionales if/else, bucles for/while/do-while).	25h
	UT3. Excepciones y errores.	Gestión de errores y excepciones básicas. Primer contacto con la Programación Orientada a Objetos: clases, objetos y métodos.	30h
Segundo Trimestre	UT 4. Programación orientada a objetos.	POO Avanzada: Herencia, polimorfismo, interfaces y clases abstractas.	40h
	UT 5. Estructuras de datos.	Estructuras de almacenamiento: Arrays y colecciones (<i>Lists, Sets, Map</i>).	25h
	UT 6. Entrada y salida de datos.	Lectura y escritura de información (Ficheros de texto, binarios y <i>XML</i>).	25h
Tercer Trimestre	UT 7. Bases de datos.	Persistencia en bases de datos (<i>IndexDB</i>) y escritura con SQL.	30h
	UT 8. UX y diseño de interfaces	Creación de interfaces gráficas e introducción a la programación de eventos.	30h
	UT 9. Proyecto final: Desarrollo de una IA de Ajedrez.	Aplicación práctica e integradora de algoritmia, árboles de búsqueda y evaluación de software mediante competición.	27h

Fuente: Elaboración propia

2.6.2. Cronograma de la Unidad de Trabajo del proyecto final IA de ajedrez

La UT 9 tiene una duración base de 27 horas lectivas, distribuidas en 9 sesiones de 3 horas cada una a lo largo de 3 semanas.

Esta temporalización está diseñada bajo un enfoque iterativo. Las sesiones combinan la investigación teórica autónoma con el desarrollo práctico de código, en las clases se explicarán distintos modos de abordar el problema de elegir la mejor jugada para la IA y los alumnos tendrán que buscar maneras de implementarlo. Durante las sesiones también se resolverán dudas surgidas por los alumnos, se proveerá feedback del código desarrollado y se le probará el código antes de la fase del torneo. En la mitad de la unidad de trabajo y en la parte final se harán sendos torneos de ajedrez en los cuales se establecerá los ratings Elo de las IAs de los alumnos, se evaluará su evolución y, en el caso del primer torneo, se proveerá feedback para mejorar el algoritmo de la IA.

A continuación, se detalla el cronograma específico de las sesiones:

Tabla 3. *Cronograma de las sesiones*

Semana	Sesión y duración	Actividades
Semana 1	Sesión 01 (3 horas)	Análisis del problema y Setup: Clonar de repositorio base de GitHub. Comprender el código proporcionado y definición de las estructuras de datos para representar la información de las piezas.
	Sesión 02 (3 horas)	Desarrollo algorítmico inicial: Diseño y programación de la función principal encargada de analizar el array de movimientos legales y elegir la jugada más ventajosa mediante cálculo de fuerza bruta básica.

	Sesión 03 (3 horas)	Primer test del código entregado, competición entre otros bots de IA de ajedrez para evaluar el Elo inicial con el que cada estudiante va a partir
Semana 2	Sesión 04 (3 horas)	Optimización e introducción de árboles de búsqueda para elegir una jugada mejor. Revisión y refactorización del código e implementación de árboles de búsqueda sencillos (prever 2-3 jugadas por delante).
	Sesión 05 (3 horas)	Testing gamificado. Verificación del funcionamiento del programa. Las IAs de los alumnos compiten entre sí de manera automatizada para generar un segundo rating Elo.
	Sesión 06 (3 horas)	Feedback del resultado de la competición entre estudiantes, revisión del código y propuestas de mejoras de los algoritmos de la IA para mejorar de cara al siguiente torneo
Semana 3	Sesión 07 (3 horas)	Revisión de la implementación del feedback, comprobar que la IA ha mejorado desde la última vez que compitió, mirar que la IA actual tiene mejor rating que la misma IA en versiones anteriores haciéndola competir consigo misma, añadir teoría de aperturas para mejorar la IA.
	Sesión 08 (3 horas)	Documentación del código y los algoritmos, preparar una pequeña presentación explicando el código para después del torneo final, acabar de depurar los errores que queden de cara al torneo final.
	Sesión 09 (3 horas)	Torneo Final y Defensa: Competición definitiva de las IAs. Cálculo del Elo final para medir el progreso. Puesta en común donde cada alumno justifica técnicamente las decisiones algorítmicas de su código.

Fuente: Elaboración propia

2.7. Selección y organización de los recursos y materiales

Para la implementación de esta situación de aprendizaje, la selección y organización de los recursos se ha estructurado de la siguiente manera:

2.7.1. Recursos físicos y espaciales

El recurso espacial principal será el aula de informática, la cual tiene equipos para que cada alumno pueda desarrollar el código. También se utilizará el aula para la explicación del proyecto, así como desarrollar la implementación de árboles de búsqueda en las sesiones posteriores. Por otro lado, este espacio será útil para que los alumnos puedan desarrollar el código con la presencia del docente y les pueda asistir en sus dudas. Por último, este espacio permitirá que al final del proyecto los alumnos puedan presentar sus IAs y explicar las decisiones de código tomadas para ellas.

2.7.2. Recursos tecnológicos y software

El entorno de desarrollo digital estará compuesto por herramientas gratuitas y estándares dentro del desarrollo de aplicaciones. Las herramientas seleccionadas son:

- Visual Studio Code: Será el entorno de desarrollo para la escritura y depuración del código.

- Navegador: Hará falta proveer un navegador (Chrome, Firefox, Safari...) para poder ejecutar y visualizar el código.
- NodeJS y JavaScript: NodeJS se utilizará como el entorno de ejecución en el lado del servidor, utilizando JavaScript como el lenguaje de programación central del proyecto.
- GitHub: Plataforma para la gestión de repositorios, el control de versiones y la publicación final del código elaborado por los estudiantes.
- Librerías y APIs específicas: Se integrarán librerías de ajedrez especializadas, como chess.js, para facilitar la validación de movimientos legales, así como el uso de plataformas externas como Lichess para la evaluación de jugadas.

2.7.3. Recursos didácticos elaborados por el docente

Para guiar el proceso de aprendizaje y que los alumnos tengan una base sobre la cual desarrollar el código, el docente actuará como facilitador aportando materiales de creación propia:

- Repositorio base: Se creará y compartirá un repositorio inicial en GitHub que contendrá el esqueleto del proyecto. Este código base incluirá la interfaz del tablero y las reglas de movimiento ya programadas, permitiendo que el desarrollo del alumno se centre exclusivamente en el reto de programar la lógica y toma de decisiones de la IA.
- Recursos audiovisuales: Se proporcionará a los estudiantes material en vídeo, elaborado por el docente, en el que se explicará paso a paso cómo clonar e iniciar el proyecto base desde GitHub, mostrando además ejemplos visuales del funcionamiento de una IA de ajedrez operando en un entorno real. Sobre este punto cabe reseñar que ya he creado un repositorio en [GitHub](#) donde los alumnos pueden tener una inicialización para empezar a escribir el código de la IA y mirar cómo funciona en un entorno real. En la bibliografía se adjunta el video explicando cómo iniciar el proyecto y haciendo algunos ejemplos de cómo sería la IA de ajedrez.

2.8. Atención a la diversidad

La elección del ajedrez para esta programación didáctica viene motivada en parte por la versatilidad que nos da a la hora de adaptar el aprendizaje, garantizando que todos los

estudiantes tengan las mismas oportunidades de progresar y aprender. Para ello, el proyecto ha tenido en cuenta los tres principios que componen el diseño universal para el aprendizaje:

1. Proporcionar múltiples formas de implicación: La actividad se presenta como un reto abierto, que carece de una solución perfecta o cerrada, se permite adaptar el desarrollo del proyecto a las necesidades individuales de cada alumno. Esta flexibilidad asegura que todos los estudiantes tengan las mismas oportunidades de aprender a aprender (Neubauer et al., 2023). La escalabilidad del proyecto facilita que el alumnado con mayores dificultades pueda crear un algoritmo más básico centrado en una evaluación de la posición y el material más sencillo, mientras que aquellos con altas capacidades pueden expandir su código implementando funciones de cálculos de profundidad y árboles de búsqueda. Asimismo, el proyecto incluye una parte de gamificación con el uso del sistema de rating Elo, lo cual fomenta la motivación y evalúa la progresión continua en las distintas fases del proyecto, permitiendo que cada estudiante avance según su propio ritmo y capacidad.
2. Proporcionar múltiples formas de representación: El proyecto permite ser abordado desde diferentes perspectivas e intereses; desde la teoría de aperturas, pasando por la evaluación de material hasta la evaluación de táctica con varias jugadas de profundidad. Además, los alumnos pueden procesar la información desde el punto de vista puramente algorítmico o desde la lógica del ajedrez, lo que beneficia a estudiantes con diferentes estilos de aprendizaje. Para apoyar el razonamiento abstracto de la programación de la IA, se utilizarán distintos ejercicios de técnica, evaluación de material y aperturas antes de trasladarlo al código. Estos ejercicios contarán con diferentes niveles de dificultad para promover, como se ha explicado en el punto anterior, distintas formas de implicación. En conjunto, se aplica una metodología de *learning by doing* que facilita la comprensión de conceptos abstractos, como las estructuras de datos, a través de la experimentación directa. Por último, se incluye también en el proyecto varias sesiones de aula invertida, que fomentan que los alumnos puedan aprender a su propio ritmo, revisitando el contenido las veces que necesiten, preparándose el material antes de la clase.
3. Proporcionar múltiples formas de evaluación: La evaluación no se limitará a una calificación final, sino que medirá el proceso a través de distintas rúbricas aplicadas a

mitad y al final del proyecto. Al comparar el Elo de la IA al concluir la práctica con el obtenido en torneos anteriores, se valorará el esfuerzo y la evolución del código del estudiante, independientemente del resultado final en la clasificación. Esto contempla casuísticas donde una IA pueda tener un Elo bajo al final del proyecto, pero demuestre una evolución técnica enorme respecto a su punto de partida. A parte del Elo, también se medirán otros elementos de la práctica, como las decisiones de diseño a la hora de programar la IA, la calidad del código y la defensa de los alumnos de su proyecto.

2.8.1. Casos específicos de atención a la diversidad

Estas medidas permitirán dar una respuesta educativa adaptada a dos casos concretos de diversidad:

- Alumna con altas capacidades (Maestra internacional): Dada su experiencia y conocimientos del ajedrez, el planteamiento abierto del proyecto le permitirá profundizar en los algoritmos de búsqueda para encontrar la mejor jugada. Podrá expandir la complejidad de su código sin limitaciones, lo cual le permitirá mejorar en sus habilidades de desarrollo de código.
- Alumno con baja alfabetización digital: Este alumno podrá elaborar una IA mucho más sencilla y funcional, adaptada a su nivel de manejo tecnológico. Gracias a las múltiples formas de evaluación, no será calificado únicamente por el nivel final de su IA o los enfrentamientos perdidos, sino que se valorará la calidad del código presentado y el esfuerzo demostrado en la evolución de su trabajo desde el primer torneo hasta el último.

2.9. Evaluación

El enfoque propuesto es una evaluación orientada al aprendizaje, donde sea una oportunidad de retroalimentación y mejora durante el proceso. Como evidencias de aprendizaje el indicador más claro será el código (evidencia de producto), se evaluará el uso correcto de variables, funciones, eficiencia de algoritmos, técnicas de clean code y escalabilidad. Los alumnos podrán explicar por qué han tomado esas decisiones a la hora de escribir la IA y que cambiarían si pudieran hacerla de cero de nuevo. Por otro lado, con la competición entre IAs se les dará un rating Elo (evidencia de ejecución) a cada IA, que les servirá como evaluación real y justa. Este rating se les dará en la sesión 5 y en la sesión 9. La evaluación no solo tendrá

en cuenta el rating Elo final, sino la evolución con el rating entre las dos sesiones. Por último, se les pedirá a los alumnos que documenten el código (evidencia documental) justificando el porqué de las funciones elegidas y el cómo funcionan.

Los criterios se derivan de los Resultados de Aprendizaje del módulo:

Tabla 4. Criterios de evaluación

Criterios de evaluación	Actividades de aprendizaje	Actividades y procedimientos de evaluación	Instrumentos	Valor en la SA
RA 3 y RA 4: Se han modificado y optimizado programas usando variables, tipos avanzados de datos (<i>arrays</i> , objetos), estructuras de repetición y expresiones lógicas complejas.	Fase de desarrollo e investigación: Creación iterativa de la función principal encargada de analizar las jugadas y evaluar material. Construcción de árboles de búsqueda y refactorización guiada por <i>feedback</i> .	Evaluación de producto: Entrega y revisión del repositorio final de código en GitHub, comprobando la estructura, legibilidad (<i>clean code</i>) y eficiencia del algoritmo.	Rúbrica técnica de evaluación de código, revisando la documentación del código aportada (Rúbrica de evaluación).	50%
RA 7: Se han probado y depurado los programas de forma autónoma, completando planes de pruebas para verificar su funcionamiento.	Testing gamificado: Torneo inicial (Sesión 5) para detectar establecer un primer nivel de Elo, torneo final para revisar el código final entregado y depuración autónoma del código con el <i>feedback</i> del primer torneo.	Evaluación de ejecución: Torneo definitivo en formato de liga suiza (Sesión 9) enfrentando a todas las IAs. Medición del desempeño técnico y robustez del software en un entorno real.	Registro automatizado del rating Elo y progresión del desempeño (evidencia de ejecución).	30%
RA 5 y Competencias Transversales: Se han reconocido componentes de POO, se han tomado decisiones fundamentadas analizando variables complejas y se asumen los riesgos de innovación.	Abstracción y debate analítico: Análisis del código entregado y justificación a través de la documentación de la lógica del código y las decisiones algorítmicas tomadas durante la programación.	Evaluación documental: Exposición breve en el aula defendiendo las decisiones técnicas tomadas y entrega de una memoria explicativa del funcionamiento de la IA diseñada.	Memoria técnica explicativa y lista de cotejo para la defensa oral (Evidencia documental).	20%

Fuente: Elaboración propia

Según los criterios descritos en la tabla anterior, se podrá aplicar la siguiente rúbrica al proyecto entregado por el alumno:

Tabla 5. Rúbrica de evaluación

Rúbrica de evaluación				
Criterio de evaluación	Muy bajo	Bajo	Medio	Alto
Aplicación de estructuras de control y algoritmos. (RA 3)	El código no se puede ejecutar o el programa no funciona como es esperado	La lógica es aleatoria o presenta errores sintácticos en las estructuras de control.	Utiliza bucles y condicionales para evaluar jugadas, pero con una estrategia simple.	Implementa un algoritmo de búsqueda complejo con lógica condicional avanzada.

Manipulación de estructuras de datos. (RA 4)	No se utilizan ninguna estructura de datos más allá de la declaración de variables	Tiene dificultades para extraer la información necesaria de las estructuras de datos proporcionadas.	Utiliza correctamente las estructuras de datos, pero con código redundante o poco optimizado.	Utiliza correctamente las estructuras de datos adecuadamente dependiendo de la situación y de manera optimizada.
Calidad y modularidad del código. (RA 3 y 4)	El código es ilegible y no permite extenderlo	Código desordenado, difícil de leer o sin respetar convenciones de estilo.	El código es funcional pero monolítico (una sola función larga) o carece de comentarios explicativos.	El código está estructurado en funciones auxiliares reutilizables, bien comentado y con nomenclatura clara.
Funcionalidad y robustez (Depuración). (Relacionado con RA 7)	El programa se bloquea o devuelve errores de ejecución durante el torneo.	El programa funciona, pero el rating es peor que una IA que elige jugadas al azar	El rating es mejor que una IA que mire 1 jugadas por delante	El rating es mejor que una IA que mire 3 jugadas por delante

Fuente: Elaboración propia

Para complementar los instrumentos de evaluación, se implementará una autoevaluación diagnóstica al inicio del proyecto. Dado que el desarrollo de una Inteligencia Artificial requiere la combinación de conocimientos de programación aprendidos durante el curso y nociones de ajedrez, es interesante hacer un estudio de nivel para poder orientar las sesiones al nivel y las necesidades de la clase.

Esta herramienta diagnóstica cumple una doble función pedagógica. Por un lado, nos permite anticiparnos a las necesidades de los estudiantes y poder reforzar el temario a impartir para reforzar el aprendizaje. Por otro lado, también nos provee un feedback de lo que aprendieron los estudiantes en el desarrollo de la práctica y darnos información relevante para repetir esta unidad de trabajo en futuros cursos.

A continuación, se detalla el modelo de cuestionario que se entregará a los estudiantes antes de la primera sesión:

Tabla 6. Rúbrica de autoevaluación diagnóstica para el alumnado

Dimensión para evaluar	Ítem de autoevaluación	Nivel de dominio (1-5)	Dificultades u observaciones del alumno
Conocimientos del ajedrez	Conozco el movimiento de todas las piezas y las reglas básicas del juego (jaque, jaque mate, enroque, etc.).		
	Entiendo el valor material de las piezas (ej. dama 9 puntos, torre 5 puntos...) y conceptos básicos de táctica (clavada, ataque doble, sobrecarga...).		
Conocimientos de programación	Entiendo y soy capaz de utilizar estructuras de control (bucles for/while, condicionales if/else) en JavaScript.		
	Comprendo cómo manipular estructuras de datos (como <i>arrays</i> bidimensionales u objetos) para almacenar información.		

	Controlo las instrucciones de entrada y salida y soy capaz de hacer operaciones de escritura y lectura de datos.		
	Conozco conceptos básicos de testing, entiendo cómo crear un test para probar mi código e interpretar los resultados del test.		
Actitud y motivación	Tengo interés por descubrir cómo funciona una inteligencia artificial.		
	Me siento capaz de afrontar un reto con una solución abierta.		

Fuente: Elaboración propia.

2.10.Situación de aprendizaje/ unidad de trabajo

Dentro de la asignatura de programación, se estructuran 9 unidades de trabajo repartidas a lo largo del año estructuradas en las siguientes unidades de trabajo:

- UT 1. Introducción.
- UT 2. Estructuras de control de código.
- UT3. Excepciones y errores.
- UT 4. Programación orientada a objetos.
- UT 5. Estructuras de datos.
- UT 6. Entrada y salida de datos.
- UT 7. Bases de datos.
- UT 8. UX y diseño de interfaces
- UT 8. Proyecto final: Desarrollo de una IA de Ajedrez.

En nuestro caso, vamos a describir la unidad de trabajo 8, que es con la que concluye la asignatura; A continuación, se detallan cada una de las sesiones que se impartirán:

2.10.1. Sesiones de la unidad de trabajo

Tabla 7. Sesión 1 del proyecto

UT8: Proyecto final: Desarrollo de una IA de ajedrez.	
ÁREA Y NIVEL: Programación - Ciclo formativo superior desarrollo de aplicaciones web	TEMPORALIZACIÓN: 3 horas
TÍTULO Y JUSTIFICACIÓN DE LA SESIÓN: Sesión 1: Introducción al proyecto.	
Sesión de inicio y explicación de la actividad, se establecerán las bases del proyecto sobre la que los alumnos trabajarán las próximas semanas.	
TAREA: Se les dará un repositorio en GitHub y una explicación con un video de como montar el proyecto. Además, se explicarán las funciones básicas que tiene ese repositorio sobre las que deberán implementar código para desarrollar la práctica. En la sesión el profesor acompañará a los alumnos para que, al terminar la clase, todos los alumnos tengan instalado el repositorio en su ordenador y puedan empezar a probar cómo funciona la IA.	
METODOLOGÍA: En esta sesión la metodología será de aula invertida. Los alumnos podrán prepararse la clase siguiendo las instrucciones desde casa y luego en clase, con la asistencia del profesor si así lo necesitasen, continuar con el proyecto.	
RESULTADOS DE APRENDIZAJE:	
• RA 4: Desarrollar programas organizando la información en estructuras de datos.	
ESTRUCTURA DE LA UNIDAD DE TRABAJO	GESTIÓN DE LOS APRENDIZAJES EN EL AULA

SECUENCIA DE ACTIVIDADES	SELECCIÓN DE EJERCICIOS	TÉCNICAS / ESTRATEGIAS METODOLÓGICAS	ORGANIZACIÓN SOCIAL + EVALUACIÓN	RECURSOS DIDÁCTICOS Y DIGITALES	ATENCIÓN A LA DIVERSIDAD
Establecer el repositorio en su ordenador individual.	• Ver la explicación del proyecto en video. • Descargarse y montar el proyecto en su ordenador.	Aula invertida.	Organización social: Trabajo individual y explicaciones en grupo clase. Evaluación: Evaluación diagnóstica y observación directa	Ordenador individual, proyector para explicar el setup, entorno de desarrollo (Visual Studio Code), GitHub (repositorio) y vídeo tutorial explicativo elaborado por el docente para el setup inicial.	Refuerzo con video para que los alumnos puedan revisualizar el contenido las veces que necesiten hasta que les quede claro.
Establecer las primeras funciones para la IA	• Escribir la primera función de selección de jugada para la IA • Testear que funciona bien ese código.	Aprendizaje basado en proyectos	Organización social: Trabajo individual (desarrollo de código) con interacción con el docente. Evaluación: Evaluación diagnóstica y observación directa	Ordenador individual, Visual Studio Code.	Se adapta el cómo escribir código y la elección de la mejor jugada al conocimiento del alumno, no habiendo un método perfecto para hacerlo.

Fuente: Elaboración propia.

Tabla 8. Sesión 2 del proyecto

UT8: Proyecto final: Desarrollo de una IA de ajedrez.					
ÁREA Y NIVEL: Programación - Ciclo formativo superior desarrollo de aplicaciones web				TEMPORALIZACIÓN:	
TÍTULO Y JUSTIFICACIÓN DE LA SESIÓN: Sesión 2: Explicación de selección de la mejor jugada.				3 horas	
Sesión para profundizar en la selección de la mejor jugada y el razonamiento de la IA.					
TAREA: Se les dará una serie de posiciones de ajedrez donde será evidente la evaluación de la mejor jugada (habrá algún mate forzado), o algún jaque. Se les pedirá resolver cada uno de los ejercicios e implementar la lógica que han utilizado para resolver los ejercicios en la IA.					
METODOLOGÍA: Se utilizará el aprendizaje basado en problemas, a partir de unos problemas los alumnos tendrán que buscar soluciones para después implementar el razonamiento de esa solución en el código de la IA. También se utilizará la metodología del aula invertida especial, donde los alumnos utilizarán lo aprendido en la primera parte de la clase y, con asistencia del profesor si así lo necesitasen, escribirán el código de la solución en su IA.					
RESULTADOS DE APRENDIZAJE:					
• RA 3: Escribir código utilizando estructuras de control.					
ESTRUCTURA DE LA UNIDAD DE TRABAJO		GESTIÓN DE LOS APRENDIZAJES EN EL AULA			
SECUENCIA DE ACTIVIDADES	SELECCIÓN DE EJERCICIOS	TÉCNICAS / ESTRATEGIAS METODOLÓGICAS	ORGANIZACIÓN SOCIAL + EVALUACIÓN	RECURSOS DIDÁCTICOS Y DIGITALES	ATENCIÓN A LA DIVERSIDAD
Explicación de la selección de la mejor jugada	• Mostrar posiciones donde exista la posibilidad de dar jaque mate • Mostrar posiciones donde haya una posibilidad de dar mate en varias jugadas.	Aprendizaje basado en problemas. Se les entregarán a los alumnos los problemas a resolver que les serán útiles a la hora de desarrollar el proyecto.	Organización social: Trabajo en pequeños grupos o toda la clase para encontrar la mejor jugada y trabajo individual para implementar la solución en la IA.	Ordenador individual, proyector para explicar las distintas posiciones.	Distintos problemas con distintas dificultades para que los alumnos con más dificultades puedan resolver los sencillos y los más aventajados puedan resolver

			Evaluación: Evaluación directa, coevaluación, heteroevaluación.		hasta el final. Lo interesante es la implementación del código.
Implementación de la selección de la mejor jugada	• Escribir el código que implemente la lógica de selección de la mejor jugada cuando hay opciones de mate. • Testear que el código funciona correctamente.	Aula invertida especial. Los alumnos implementarán la solución del código ayudándose del profesor para resolver dudas.	Organización social: Trabajo individual (desarrollo de código) con interacción con el docente. Evaluación: Evaluación directa, pruebas de ejecución y autoevaluación.	Ordenador individual, Visual Studio Code.	Se adapta el cómo escribir código y la elección de la mejor jugada al razonamiento que usaron los alumnos en el problema anterior.

Fuente: Elaboración propia.

Tabla 9. Sesión 3 del proyecto.

UT8: Proyecto final: Desarrollo de una IA de ajedrez.					
ÁREA Y NIVEL: Programación - Ciclo formativo superior desarrollo de aplicaciones web				TEMPORALIZACIÓN: 3 horas	
TÍTULO Y JUSTIFICACIÓN DE LA SESIÓN: Sesión 3: Desarrollo de código, buscar la mejor jugada.					
Sesión para profundizar en la selección de la mejor jugada y el razonamiento de la IA.					
TAREA: De forma similar a la sesión anterior, esta vez se les dará una serie de posiciones donde también será evidente la selección de la mejor jugada, pero en este caso, la solución será algún tipo de ventaja ya sea forzar jugada (jaques y amenazas) o ganar material (ganar una pieza, por ejemplo).					
METODOLOGÍA: Se utilizará el aprendizaje basado en problemas, a partir de unos problemas los alumnos tendrán que buscar soluciones para después implementar el razonamiento de esa solución en el código de la IA. También se utilizará la metodología del aula invertida especial, donde los alumnos utilizarán lo aprendido en la primera parte de la clase y, con asistencia del profesor si así lo necesitasen, escribirán el código de la solución en su IA.					
RESULTADOS DE APRENDIZAJE:					
• RA 3: Escribir código utilizando estructuras de control. • RA 4: Desarrollar programas organizando la información en estructuras de datos.					
ESTRUCTURA DE LA UNIDAD DE TRABAJO		GESTIÓN DE LOS APRENDIZAJES EN EL AULA			
SECUENCIA DE ACTIVIDADES	SELECCIÓN DE EJERCICIOS	TÉCNICAS / ESTRATEGIAS METODOLÓGICAS	ORGANIZACIÓN SOCIAL + EVALUACIÓN	RECURSOS DIDÁCTICOS Y DIGITALES	ATENCIÓN A LA DIVERSIDAD
Explicación de la selección de la mejor jugada	• Mostrar posiciones donde exista de dar jaque para amenazar algo • Mostrar posiciones donde haya una posibilidad de ganar una pieza. • Mostrar posiciones donde haya posibilidad de ganar material.	Aprendizaje basado en problemas. Se les entregarán a los alumnos los problemas a resolver que les serán útiles a la hora de desarrollar el proyecto.	Organización social: Trabajo en pequeños grupos o toda la clase para encontrar la mejor jugada y trabajo individual para implementar la solución en la IA. Evaluación: Evaluación directa, coevaluación, heteroevaluación.	Ordenador individual, proyector para explicar las distintas posiciones.	Distintos problemas con distintas dificultades para que los alumnos con más dificultades puedan resolver los sencillos y los más aventajados puedan resolver hasta el final. Lo interesante es la implementación del código.
Implementación de la selección de la mejor jugada	• Escribir el código que implemente la lógica de selección de la mejor jugada cuando hay opciones de mate.	Aula invertida especial. Los alumnos implementarán la solución del código	Organización social: Trabajo individual (desarrollo de código) con interacción con el docente.	Ordenador individual, Visual Studio Code.	Se adapta el cómo escribir código y la elección de la mejor jugada al razonamiento que usaron los alumnos

	• Testear que el código funciona correctamente.	ayudándose del profesor para resolver dudas.	Evaluación: Evaluación directa, pruebas de ejecución y autoevaluación.		en el problema anterior.
--	---	--	--	--	--------------------------

Fuente: Elaboración propia.

Tabla 10. Sesión 4 del proyecto

UT8: Proyecto final: Desarrollo de una IA de ajedrez.					
ÁREA Y NIVEL: Programación - Ciclo formativo superior desarrollo de aplicaciones web				TEMPORALIZACIÓN: 3 horas	
TÍTULO Y JUSTIFICACIÓN DE LA SESIÓN: Sesión 4: Desarrollo de código, árboles de búsqueda.					
Sesión para profundizar iterar sobre el código que ya tienen los alumnos para añadir árboles de búsqueda a la solución que ya tienen.					
TAREA: En esta sesión se analizará lo ya aprendido y con esa lógica se implementará una función que permita evaluar la posición en base al material y a las opciones de mate que tiene la posición después de N jugadas.					
METODOLOGÍA: Aula invertida especial. El profesor dará unas pautas para elaborar una función de evaluación de material y otra de evaluación de posición. Una vez estén explicadas las funciones, los alumnos deberán de implementarlas en su proyecto con la asistencia del profesor si así lo necesitasen.					
RESULTADOS DE APRENDIZAJE:					
- RA 3: Escribir código utilizando estructuras de control. - RA 5: Desarrollar programas aplicando características de la programación orientada a objetos.					
ESTRUCTURA DE LA UNIDAD DE TRABAJO		GESTIÓN DE LOS APRENDIZAJES EN EL AULA			
SECUENCIA DE ACTIVIDADES	SELECCIÓN DE EJERCICIOS	TÉCNICAS / ESTRATEGIAS METODOLÓGICAS	ORGANIZACIÓN SOCIAL + EVALUACIÓN	RECURSOS DIDÁCTICOS Y DIGITALES	ATENCIÓN A LA DIVERSIDAD
Explicación de cómo evaluar el material y la posición.	- Mostrar posiciones donde exista desigualdad de material para evaluar qué bando está mejor. - Mostrar posiciones donde haya desigualdad de material, pero opciones de mate para cambiar esa evaluación.	Aprendizaje basado en problemas. Se les entregarán a los alumnos los problemas a resolver que les serán útiles a la hora de desarrollar el proyecto.	Organización social: Trabajo en pequeños grupos o toda la clase para evaluar el material y la posición. Evaluación: Evaluación directa, heteroevaluación, coevaluación.	Ordenador individual, proyector para explicar las distintas posiciones.	Distintos problemas con distintas dificultades para que los alumnos con más dificultades puedan resolver los sencillos y los más aventajados puedan resolver hasta el final. Lo interesante es la implementación del código.
Implementación de funciones de evaluación de material y evaluación de posición	- Hacer un ejercicio grupal para implementar el razonamiento de los ejercicios anteriores en una estructura de código que contemple todos los casos. - Explicar cómo iterar sobre esas funciones para poder elegir la mejor jugada en base a una profundidad N del código.	Aula invertida especial. Los alumnos de manera grupal darán ideas de cómo implementar la solución y el guiará las ideas para implementar la solución grupal.	Organización social: Toda la clase para dar ideas de cómo implementar el código. Evaluación: Evaluación directa, coevaluación.	Proyector para mostrar cómo implementar la solución.	Al participar todos los alumnos entre ellos y darse ideas mutuamente, los alumnos más avanzados podrán ayudar a los alumnos con más dificultades explicando cómo funciona la solución. También está adaptado porque todos los alumnos pueden aportar ideas, independientemente del nivel.

Implementación de la evaluación del material y la posición.	• Escribir el código explicado por el profesor en la IA de cada alumno. • Testear que el código funciona correctamente.	Aula invertida especial. Los alumnos implementarán la solución del código ayudándose del profesor para resolver dudas.	Organización social: Trabajo individual (desarrollo de código) con interacción con el docente.	Ordenador individual, Visual Studio Code.	Se adapta el cómo escribir código y la implementación al razonamiento que usaron los alumnos en la lluvia de ideas.
---	--	--	--	---	---

Fuente: Elaboración propia.

Tabla 11. Sesión 5 del proyecto

UT8: Proyecto final: Desarrollo de una IA de ajedrez.					
ÁREA Y NIVEL: Programación - Ciclo formativo superior desarrollo de aplicaciones web				TEMPORALIZACIÓN:	
TÍTULO Y JUSTIFICACIÓN DE LA SESIÓN: Sesión 5: Torneo de evaluación gamificada.				3 horas	
Primera sesión de evaluación de las IAs de los estudiantes para evaluar el código actual y sacar propuestas de mejora.					
TAREA: En esta sesión se creará un entorno donde poder poner a competir las IAs de los alumnos entre ellas, después de que compitan todas contra todas se les asignará un rating Elo. Ese será el rating inicial de cada IA de cada alumno, al final del proyecto volverán a competir todas las IAs y se podrá medir la progresión de las siguientes sesiones.					
METODOLOGÍA: Gamificación y aula invertida especial. El profesor dará un código y unas IAs ya evaluadas para que los alumnos puedan probar su IA contra estas otras IAs. Una vez implementada esa función, los alumnos tendrán que almacenar la salida de la misma en una tabla Excel y calcular el rating Elo del torneo.					
RESULTADOS DE APRENDIZAJE:					
- RA 4: Desarrollar programas organizando la información en estructuras de datos. - RA 7: Verificar el funcionamiento de los programas diseñando y realizando pruebas.					
ESTRUCTURA DE LA UNIDAD DE TRABAJO		GESTIÓN DE LOS APRENDIZAJES EN EL AULA			
SECUENCIA DE ACTIVIDADES	SELECCIÓN DE EJERCICIOS	TÉCNICAS / ESTRATEGIAS METODOLÓGICAS	ORGANIZACIÓN SOCIAL + EVALUACIÓN	RECURSOS DIDÁCTICOS Y DIGITALES	ATENCIÓN A LA DIVERSIDAD
Explicación del código para que compitan las IAs entre ellas.	- Mostrar el código de distintas IAs de ejemplo para que los alumnos las tengan de referencia para competir. - Mostrar el código de cómo hacer competir varias partidas a 2 IAs.	Aula invertida, los alumnos podrán tener esta información desde antes de la clase en un repositorio y consultarla antes de empezar, y utilizar la clase para resolver dudas.	Organización social: Trabajo individual para establecer el código en el ordenador de cada alumno. Evaluación: Evaluación directa, heteroevaluación.	Ordenador individual, proyector para explicar las distintas posiciones.	Los alumnos podrán disponer del código de antemano y revisarlo antes de la clase las veces que necesiten, adaptándose al ritmo del alumno.
Ejecutar torneo entre las IAs de prueba y las IAs de otros compañeros	- Ejecutar el código de competir entre IAs entre las IAs de prueba y las del alumno. - Ejecutar el código de competir entre IAs entre las IAs de prueba y las de otros compañeros. - Guardar la ejecución de las competiciones en una hoja Excel para poder calcular el Elo, y revisar las partidas para	Gamificación. Los alumnos podrán evaluar el nivel de su IA haciéndola competir contra otras IAs.	Organización social: En grupos de 2 para hacer competir las IAs, e individual para hacer competir entre las IAs de prueba. Evaluación: Evaluación directa, pruebas de ejecución, evidencia de producto (hoja de Excel), heteroevaluación y coevaluación.	Ordenador individual, visual Studio Code.	Independientemente del resultado de la competición, será solo una prueba de nivel para comprobar la mejora de los cambios que implemente el alumno a lo largo de las próximas sesiones. Esto permite que los distintos ritmos de aprendizaje se reflejen por igual en

	estudiar propuestas de mejora				la evolución de la IA.
--	-------------------------------	--	--	--	------------------------

Fuente: Elaboración propia.

Tabla 12. Sesión 6 del proyecto

UT8: Proyecto final: Desarrollo de una IA de ajedrez.					
ÁREA Y NIVEL: Programación - Ciclo formativo superior desarrollo de aplicaciones web TÍTULO Y JUSTIFICACIÓN DE LA SESIÓN: Sesión 6: Evaluación de código y propuestas de mejora. Sesión para evaluar el resultado de la competición y hacer propuestas de mejoras para la IA de cada alumno.				TEMPORALIZACIÓN: 3 horas	
TAREA: En esta sesión se analizarán algunas partidas jugadas en el torneo para evaluar qué pudo fallar, se propondrá hacer cambios en la profundidad de las IAs para ver si mejoran volver a iterar sobre la evaluación de posiciones para ver cómo poder mejorarla. También se escribirá un test que haga competir los cambios de la nueva IA con la antigua para poder analizar si los cambios la hacen mejorar o empeorar.					
METODOLOGÍA: Aula invertida especial. El profesor mostrará algunas partidas que se disputaron en el torneo para revisar cómo evaluaron las jugadas y tomará una IA de ejemplo para proponer mejoras en el código de evaluación de posición y material.					
RESULTADOS DE APRENDIZAJE: - RA 3: Escribir código utilizando estructuras de control. - RA 4: Desarrollar programas organizando la información en estructuras de datos. - RA 7: Verificar el funcionamiento de los programas diseñando y realizando pruebas.					
ESTRUCTURA DE LA UNIDAD DE TRABAJO		GESTIÓN DE LOS APRENDIZAJES EN EL AULA			
SECUENCIA DE ACTIVIDADES	SELECCIÓN DE EJERCICIOS	TÉCNICAS / ESTRATEGIAS METODOLÓGICAS	ORGANIZACIÓN SOCIAL + EVALUACIÓN	RECURSOS DIDÁCTICOS Y DIGITALES	ATENCIÓN A LA DIVERSIDAD
Explicación de las partidas de la competición.	- Mostrar partidas donde la jugada realizada no fuese la mejor y hubiera otra que fuese mejor. - Debatir en la clase como hacer esa mejora de evaluación en el código e implementarla con la colaboración de la clase.	Aprendizaje basado en problemas.	Organización social: Trabajo grupal de toda la clase para evaluar las partidas y las propuestas de mejora. Evaluación: Evaluación directa, heteroevaluación, coevaluación.	Ordenador individual, proyector para explicar las distintas partidas.	Se mostrarán partidas más sencillas de seguir donde la propuesta de mejora sea sencilla y partidas más complejas de evaluar para que la propuesta de mejora sea más difícil de analizar. Así todos los alumnos pueden participar independientemente de su nivel.
Implementación de mejora de funciones de evaluación de material y evaluación de posición	- Realizar el mismo ejercicio de revisión de partidas con las partidas que ha jugado la IA del alumno. - Implementar las mejoras de código para las partidas analizadas. - Escribir un test que permita revisar después de los cambios si la IA tiene más o menos	Aula invertida especial. Los alumnos implementarán las mejoras con asistencia del profesor si así lo necesitasen. Learning by doing.	Organización social: individual. Evaluación: Evaluación directa, autoevaluación.	Proyector para mostrar cómo implementar la solución.	Al no haber un método único de realizar mejoras, las mejoras del código se adaptan al conocimiento del alumno, dejando al profesor la evaluación de la mejora de la IA y en caso de que el alumno no pudiera conseguir ninguna mejora, permitiendo evaluar el uso del

	Elo que antes de los cambios.				test para revisar si la IA mejora o no.
--	-------------------------------	--	--	--	---

Fuente: Elaboración propia.

Tabla 13. Sesión 7 del proyecto

UT8: Proyecto final: Desarrollo de una IA de ajedrez.					
ÁREA Y NIVEL: Programación - Ciclo formativo superior desarrollo de aplicaciones web				TEMPORALIZACIÓN:	
TÍTULO Y JUSTIFICACIÓN DE LA SESIÓN: Sesión 7: Implementación de las mejoras.				3 horas	
Sesión para acabar de mejorar la IA de cara al torneo final.					
TAREA: En esta sesión se continuará con la implementación de las mejoras en la evaluación de posiciones. También se explicará teoría de aperturas para dar otro enfoque de cara a mejorar la IA.					
METODOLOGÍA: Se utilizará el aprendizaje basado en problemas, los alumnos recibirán una serie de aperturas y deberán de buscar el modo de implementarlas en su IA. También se utilizará la metodología del aula invertida especial, donde los alumnos utilizarán lo aprendido en la primera parte de la clase y, con asistencia del profesor si así lo necesitasen, escribirán el código de la solución en su IA.					
RESULTADOS DE APRENDIZAJE:					
- RA 3: Escribir código utilizando estructuras de control. - RA 4: Desarrollar programas organizando la información en estructuras de datos. - RA 5: Desarrollar programas aplicando características de la programación orientada a objetos. - RA 7: Verificar el funcionamiento de los programas diseñando y realizando pruebas.					
ESTRUCTURA DE LA UNIDAD DE TRABAJO		GESTIÓN DE LOS APRENDIZAJES EN EL AULA			
SECUENCIA DE ACTIVIDADES	SELECCIÓN DE EJERCICIOS	TÉCNICAS / ESTRATEGIAS METODOLÓGICAS	ORGANIZACIÓN SOCIAL + EVALUACIÓN	RECURSOS DIDÁCTICOS Y DIGITALES	ATENCIÓN A LA DIVERSIDAD
Teoría de aperturas	- Mostrar varias aperturas y analizar por qué pueden sacar ventaja. - Escribir el código para que en el inicio de la partida la IA elija la mejor jugada basándose en una apertura. - Extender el código para que coja la información de una base de datos de aperturas.	Aprendizaje basado en problemas. Se les entregarán a los alumnos las aperturas a analizar que les serán útiles a la hora de desarrollar el proyecto.	Organización social: Trabajo en pequeños grupos o toda la clase para analizar las aperturas y trabajo individual para implementar la solución en la IA. Evaluación: Evaluación directa, coevaluación, heteroevaluación.	Ordenador individual, proyector para explicar las distintas posiciones.	Distintas aperturas con distintas dificultades para que los alumnos con más dificultades puedan resolver los sencillos y los más aventajados puedan resolver hasta el final. Lo interesante es la implementación del código.
Implementación de las aperturas	- Escribir el código que implemente la lógica de selección de una jugada de apertura o la mejor jugada dependiendo del estado de la partida. - Testear que el código funciona correctamente.	Aula invertida especial. Los alumnos implementarán la solución del código ayudándose del profesor para resolver dudas.	Organización social: Trabajo individual (desarrollo de código) con interacción con el docente. Evaluación: Evaluación directa, pruebas de ejecución y autoevaluación.	Ordenador individual, Visual Studio Code.	La cantidad de aperturas que la IA va a soportar y el tipo de aperturas se proveerán dependiendo de las necesidades del alumnado, entregando muchas aperturas tiene poco rating y viceversa para ayudar a mejorar más el nivel de la IA.

Fuente: Elaboración propia.

Tabla 14. Sesión 8 del proyecto

UT8: Proyecto final: Desarrollo de una IA de ajedrez.					
ÁREA Y NIVEL: Programación - Ciclo formativo superior desarrollo de aplicaciones web				TEMPORALIZACIÓN: 3 horas	
TÍTULO Y JUSTIFICACIÓN DE LA SESIÓN: Sesión 8: Documentación del código y preparación de las presentaciones.					
Sesión para preparar la documentación y una presentación a modo de defensa del proyecto.					
TAREA: En esta sesión los alumnos acabarán de programar su IA, documentarán el código y prepararán una presentación defendiendo las decisiones de diseño tomadas.					
METODOLOGÍA: Aprendizaje basado en proyectos, trabajo autónomo. Los alumnos ya están en la fase final del proyecto y, después de todo el recorrido, deberán de documentar el trabajo realizado, argumentar las decisiones tomadas y, posteriormente, defenderlo.					
RESULTADOS DE APRENDIZAJE:					
- RA 4: Desarrollar programas organizando la información en estructuras de datos. - RA 7: Verificar el funcionamiento de los programas diseñando y realizando pruebas.					
ESTRUCTURA DE LA UNIDAD DE TRABAJO			GESTIÓN DE LOS APRENDIZAJES EN EL AULA		
SECUENCIA DE ACTIVIDADES	SELECCIÓN DE EJERCICIOS	TÉCNICAS / ESTRATEGIAS METODOLÓGICAS	ORGANIZACIÓN SOCIAL + EVALUACIÓN	RECURSOS DIDÁCTICOS Y DIGITALES	ATENCIÓN A LA DIVERSIDAD
Testear que la IA funciona mejor que el último torneo.	- Utilizar los tests creados anteriormente para revisar que la IA funciona mejor que en el último torneo. - Acabar de realizar pruebas de código para elegir la versión de la IA con más rating.	Aprendizaje basado en problemas. Los alumnos deberán de realizar tests a su propia IA para saber cuál es la mejor versión.	Organización social: Trabajo individual (desarrollo de código) con interacción con el docente. Evaluación: Evaluación directa, autoevaluación.	Ordenador individual, proyector para explicar las distintas posiciones.	La mejora del rating de la IA siempre estará presente independientemente del nivel de esta, ya que se ha iterado para mejorarla. La diferencia entre los alumnos será cuánto habrá mejorado, pero todos los alumnos podrán realizar el ejercicio independientemente de su nivel
Documentación del proyecto.	- Documentar el código de la IA. - Crear una presentación explicando el código, las decisiones tomadas y el resultado de la IA.	Aprendizaje basado en proyectos. Los alumnos aprenden a partir del proyecto realizado.	Organización social: Trabajo individual (creación de la presentación) con interacción con el docente. Evaluación: heteroevaluación.	Ordenador individual, Visual Studio Code.	La diversidad en esta actividad se ve reflejada en que durante todas las actividades anteriores el nivel se ha adaptado a las necesidades de cada alumno, ahora para documentar el proceso y mostrar el resultado todos los alumnos serán capaces de realizarla.

Fuente: Elaboración propia.

Tabla 15. Sesión 9 del proyecto

UT8: Proyecto final: Desarrollo de una IA de ajedrez.	
ÁREA Y NIVEL: Programación - Ciclo formativo superior desarrollo de aplicaciones web TÍTULO Y JUSTIFICACIÓN DE LA SESIÓN: Sesión 9: Torneo de evaluación final y presentación de la documentación.	TEMPORALIZACIÓN: 3 horas

Sesión de evaluación del proyecto.					
TAREA: En esta sesión se evaluará el proyecto presentado por los alumnos a partir de dos puntos, la presentación y calidad del código, y el desempeño y calidad de la IA.					
METODOLOGÍA: Gamificación y aprendizaje basado en proyectos. El profesor evaluará el desempeño de la IA a través de un torneo enfrentándola a otras IAs de compañeros, con este nuevo rating se podrá evaluar la evolución del proyecto. Por otro lado, también se evaluará la calidad del código presentado y las decisiones realizadas para realizar el proyecto a través de la defensa de la presentación de los alumnos.					
RESULTADOS DE APRENDIZAJE:					
RA 7: Verificar el funcionamiento de los programas diseñando y realizando pruebas.					
ESTRUCTURA DE LA UNIDAD DE TRABAJO		GESTIÓN DE LOS APRENDIZAJES EN EL AULA			
SECUENCIA DE ACTIVIDADES	SELECCIÓN DE EJERCICIOS	TÉCNICAS / ESTRATEGIAS METODOLÓGICAS	ORGANIZACIÓN SOCIAL + EVALUACIÓN	RECURSOS DIDÁCTICOS Y DIGITALES	ATENCIÓN A LA DIVERSIDAD
Ejecutar torneo entre las IAs de los alumnos.	- Ejecutar el código para competir entre IAs de todos los alumnos. - Guardar el resultado del cambio de Elo y utilizar la rúbrica de evaluación para evaluar el resultado.	Gamificación. Los alumnos podrán evaluar el nivel de su IA haciéndola competir contra otras IAs. Aprendizaje basado en proyectos, en esta sesión se evalúa todo el trabajo realizado por los alumnos.	Organización social: Individual, cada alumno tendrá un resultado con el rating de su IA. Evaluación: Evaluación directa, pruebas de ejecución, evidencia de producto (nuevo rating Elo), rúbrica de evaluación y heteroevaluación.	Ordenador individual, visual Studio Code.	Aquí la rúbrica se adapta a la evolución de la IA durante el proyecto lo cual hace que se adapte al aprendizaje del estudiante en vez del resultado del proyecto.
Defensa de las presentaciones.	- Defender las decisiones tomadas a la hora de desarrollar la IA. - Explicar el código de la IA, su funcionamiento y mantenimiento.	Aprendizaje basado en proyectos, trabajo autónomo.	Organización social: Individual, cada alumno defenderá su proyecto. Evaluación: Evaluación directa, pruebas de ejecución, evidencia de producto (nuevo rating Elo), rúbrica de evaluación y heteroevaluación.	Proyector para las presentaciones	Aquí la rúbrica se adapta a las decisiones tomadas durante el proyecto, haciendo que el resultado sea favorable habiendo tomado buenas decisiones.

Fuente: Elaboración propia.

De este modo quedarían descritas todas las sesiones de la unidad de trabajo, a modo de resumen, se detallan en la siguiente tabla la relación de la evaluación de la unidad de trabajo con los resultados de aprendizaje:

Tabla 16. Evaluación de los resultados de aprendizaje de la unidad de trabajo.

Área: Programación		Tarea: Desarrollo de una IA de ajedrez			
Contenido	Resultados de aprendizaje	Criterios de evaluación	Competencias profesionales	Sesiones	Instrumentos
Identificación de los	Reconoce la estructura de un	Se ha modificado el código de un	Integrar contenidos en la lógica de una	S1, S3, S5, S6, S7, S8	Rúbrica de evaluación.

elementos de un programa informático	programa informático, identificando y relacionando los elementos	programa para crear y utilizar variables Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas	aplicación web, desarrollando componentes de acceso a datos adecuados a las especificaciones.		
Utilización de objetos.	- Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos. - Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje	- Se han escrito programas simples. - Se han incorporado y utilizado librerías de objetos. - Se han utilizado estructuras de repetición - Se han probado y depurado los programas	Desarrollar servicios para integrar sus funciones en otras aplicaciones web, asegurando su funcionalidad.	S5, S6, S7, S8, S9	Memoria técnica explicativa y lista de cotejo para la defensa oral.
Uso de estructuras de control, desarrollo de clases	- Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos. - Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases	- Se han reconocido la sintaxis, estructura y componentes típicos de una clase. - Se han creado y utilizado conjuntos y librerías de clases. - Se han aplicado formatos en la visualización de la información.	Completar planes de pruebas verificando el funcionamiento de los componentes software, desarrollados, según las especificaciones. Integrar servicios y contenidos distribuidos en aplicaciones web, asegurando su funcionalidad.	S4, S7	Memoria técnica explicativa y lista de cotejo para la defensa oral.
Lectura y escritura de información Aplicación de las estructuras de almacenamie nto	- Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos. - Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.	- Se han utilizado iteradores para recorrer los elementos de las listas. - Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto. - Se han realizado programas que implementen y utilicen jerarquías de clases.	Resolver situaciones, problemas o contingencias, con iniciativa y autonomía en el ámbito de su competencia, con creatividad, innovación y espíritu de mejora en el trabajo personal y en el de los miembros del equipo.	S2, S3, S4, S6, S7	Rúbrica de evaluación.

Fuente: Elaboración propia.

2.11. Evaluación de la Programación didáctica

Como documento vivo y flexible, la programación didáctica no termina con su diseño, sino que se somete a una revisión crítica para evaluar los resultados obtenidos tras su implementación y, en consecuencia, introducir las modificaciones y mejoras pertinentes para futuras ediciones.

Con el objetivo de analizar la eficacia y la viabilidad de la situación de aprendizaje planteada, de forma estructurada, se emplea una matriz DAFO, una herramienta que permite evaluar el diseño de la programación diferenciando entre las cuestiones internas que dependen del propio diseño docente (Fortalezas y Debilidades) y las cuestiones externas que dependen del entorno (Oportunidades y Amenazas).

2.11.1. Análisis DAFO de la situación de aprendizaje

- Fortalezas:
 - Innovación metodológica: La integración del aprendizaje basado en proyectos, con la gamificación y el ajedrez potencia la motivación intrínseca del alumnado. Además, el ajedrez provee una herramienta transversal entre todas las asignaturas del grado que puede servir como elemento de anclaje para el alumnado.
 - Evaluación auténtica: El diseño de la programación aporta instrumentos de evaluación basados en evidencias de ejecución reales, justas y objetivas (el rendimiento del código, el rating Elo...)
- Debilidades:
 - Carga de trabajo docente: El diseño, preparación del repositorio base en *GitHub*, soporte técnico continuado y corrección de las diversas inteligencias artificiales suponen una carga de trabajo extra para el profesorado, así como necesitar de conocimiento teórico del ajedrez.
 - Bloqueo por falta de conocimiento: El utilizar el ajedrez es una herramienta creativa muy útil, pero por otro lado tiene la contrapartida de que el alumnado necesita conocer las reglas del juego y tener un mínimo de conocimiento de técnica y teoría para poder implementar la mejor jugada para la IA. Esto puede

suponer una barrera para los alumnos que no posean este conocimiento o alumnos que les cueste más aprender teoría de juegos.

- Oportunidades:
 - Relevancia del contexto tecnológico: El auge social y laboral de la inteligencia artificial conecta el aprendizaje directamente con la realidad tecnológica actual, resultando muy pedagógico para los estudiantes.
 - Apertura al entorno y transversalidad: La naturaleza del proyecto permite la apertura del centro al entorno social, disputando torneo de ajedrez con IA. También posibilita el trabajo interdisciplinar con otros módulos, como despliegue de aplicaciones o bases de datos.
- Amenazas:
 - Reticencias estructurales: La falta de coordinación docente a la hora de alinearse para impartir el proyecto puede verse mermada por la falta de conocimiento de ajedrez o de la lógica de creación de una IA. Además, el profesorado deberá poseer conocimientos de las librerías que se utilizan para desarrollar la práctica lo cual puede suponer un desgaste extra para la docencia.
 - Baja calidad de las IAs: Al darse la posibilidad de que los alumnos tengan poco conocimiento del ajedrez y como la práctica puede adaptarse a un nivel bajo, existe la posibilidad de que los alumnos puedan entregar proyectos de baja calidad pero que cumplan con los requisitos para aprobar. Esto haría que todo el esfuerzo que se ha puesto en fomentar la innovación cayese en saco roto al no haberse utilizado de manera correcta.

Figura 3. Gráfico análisis DAFO.



Fuente: Elaboración propia.

2.11.2. Instrumentos de Autoevaluación Docente

Para enriquecer este proceso de evaluación y hacerlo cuantificable, se ha diseñado como herramienta complementaria una diana de autoevaluación docente. Tras la finalización de la situación de aprendizaje, el profesorado puntuará en esta diana gráfica (en una escala del 1 al 5) diferentes indicadores del diseño, basados en la matriz DAFO descrita en el apartado anterior:

1. Adecuación de la temporalización
2. Eficacia de los recursos digitales seleccionados
3. Grado de consecución de los resultados de aprendizaje
4. Éxito en la atención a la diversidad
5. Impacto de la gamificación
6. Idoneidad de la evaluación
7. Dificultades en la implementación del proyecto

Esta herramienta visual permitirá detectar de un vistazo las áreas de la programación que han funcionado correctamente y aquellas que requerirán una reformulación metodológica para el siguiente curso académico.

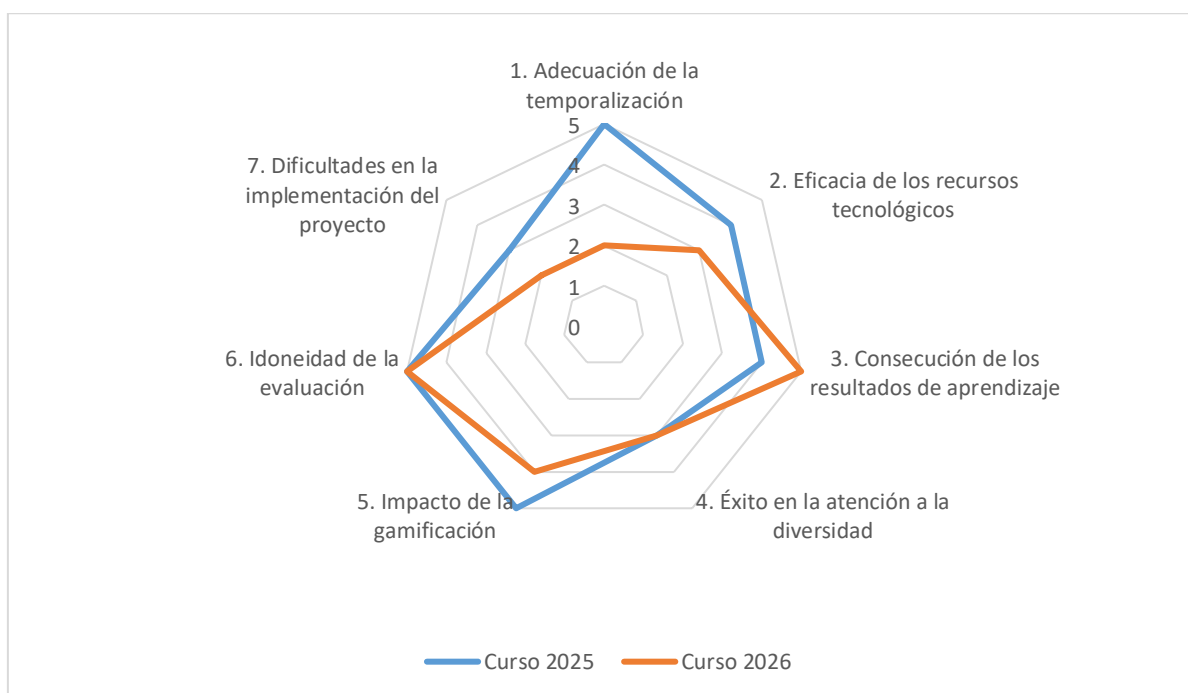
Para el último punto (dificultades en la implementación) será interesante también recoger la experiencia tanto del alumnado como del profesorado a la hora de implementar el proyecto, especialmente de cara al conocimiento previo de ajedrez por parte del alumnado y profesorado. Esto permitirá diseñar en el futuro algún tipo de curso o taller para preparar a los alumnos a esta actividad.

Tabla 17. *Elaboración de autoevaluación docente*

Indicadores docentes	Puntuación (1-5)
1. Adecuación de la temporalización	
2. Eficacia de los recursos tecnológicos	
3. Consecución de los resultados de aprendizaje	
4. Éxito en la atención a la diversidad	
5. Impacto de la gamificación	
6. Idoneidad de la evaluación	
7. Dificultades en la implementación del proyecto	

Fuente: Elaboración propia

Figura 4. *Ejemplo diana de autoevaluación*



Fuente: Elaboración propia

3. Conclusiones

Este trabajo cumple con las metas planteadas en su inicio, demostrando la viabilidad de integrar enfoques pedagógicos innovadores en la formación profesional. A través de este documento, se ha desglosado una unidad de trabajo que combina la creación de una inteligencia artificial con el uso de ajedrez a través del aprendizaje basado en proyectos, cumplido el objetivo general. Dicho diseño se ha desarrollado mediante una situación de aprendizaje innovadora que ha permitido facilitar la asimilación de fundamentos algorítmicos complejos y potenciar el pensamiento computacional del alumnado.

En relación con el primer objetivo específico, se ha analizado e implementado la fundamentación del aprendizaje basado en proyectos. La programación desarrollada muestra que esta metodología permite situar al estudiante en un rol activo y responsable, para fomentar la autonomía mediante la resolución de un problema real. Al desglosar el proyecto en nueve sesiones iterativas, se evidencia que el aprendizaje basado en proyectos fomenta un aprendizaje significativo alineado con las demandas del sector del desarrollo de software.

Respecto al segundo objetivo, centrado en evaluar el potencial pedagógico del ajedrez como herramienta transversal, el trabajo confirma su inmenso valor para el desarrollo del pensamiento computacional. La abstracción del tablero y las piezas a código ha servido como un sistema de beneficio dual, por un lado, para entrenar habilidades cognitivas transferibles a la programación, beneficiando a los estudiantes de habilidades como el reconocimiento de patrones y la toma de decisiones bajo restricciones. Y por otro lado beneficiando conceptos abstractos de programación a través del ajedrez como son el uso de arrays para elegir la mejor jugada, o concepto de árbol de búsqueda que se entiende mejor utilizando como referencia posiciones de ajedrez.

En cuanto al tercer objetivo, enfocado en explorar los conceptos fundamentales de la inteligencia artificial como vehículo de alfabetización digital, el diseño de la unidad de trabajo ha cumplido con la premisa. Al requerir que los alumnos programen su propia IA, los alumnos han transitado de consumidores pasivos a creadores activos, comprobando que la IA no es una herramienta que aprenda de la nada, sino que adquiere conocimiento a través de iteraciones del código y que los resultados que devuelve pueden ser erróneos. Esta

aproximación no solo enseña a utilizar la IA, sino que educa sobre sus mecanismos subyacentes, favoreciendo una alfabetización digital profunda.

Las principales aportaciones de este trabajo radican en ofrecer un recurso educativo plenamente adaptable a la diversidad del aula, gracias, entre otras cosas, a sistemas de gamificación, como la evaluación automatizada mediante el rating Elo. Esta propuesta aporta al ámbito de la formación profesional un modelo de evaluación auténtica que priorizando evidencias de producto y de ejecución fomentan la motivación intrínseca del alumnado.

En definitiva, este trabajo muestra que al transformar el aula en un espacio de creación tecnológica y fusionar la riqueza del ajedrez con la innovación de la inteligencia artificial, no solo se instruye a futuros desarrolladores de software, sino que se profundizan en herramientas que les serán de utilidad al terminar el grado.

4. Limitaciones y prospectiva

Durante la elaboración de este trabajo, me he enfrentado a diversas limitaciones que han condicionado su desarrollo y enfoque.

A nivel metodológico y de búsqueda de información, existe una notable escasez de fuentes académicas y bibliografía específica que aborden la enseñanza de la inteligencia artificial, debido a lo novedoso de la herramienta, así como de implementación del ajedrez como herramienta educativa. Esto ha hecho que buscar documentación para combinar ambas ideas haya sido difícil, por lo que he tenido que buscar fuentes académicas que implementasen el ajedrez y bibliografía sobre la inteligencia artificial de manera separada. Si bien abunda la literatura sobre el aprendizaje basado en proyectos y metodologías activas, la intersección concreta de la IA, el ajedrez y la programación web es un campo muy reciente y poco explorado a nivel pedagógico, lo que ha supuesto un desafío para fundamentar teóricamente el trabajo.

En segundo lugar, la limitación más significativa radica en la naturaleza hipotética de la propuesta. Al tratarse de un diseño de programación didáctica, no he tenido la posibilidad de implementarlo en un aula con un grupo real de estudiantes. Esta carencia práctica me impide comprobar de forma empírica aspectos como la viabilidad exacta de la temporalización de las nueve sesiones, la respuesta del alumnado frente al desarrollo del proyecto, o la efectividad real de las adaptaciones propuestas.

A partir de estas limitaciones, se abren diversas líneas de prospectiva y trabajo futuro. Mi aspiración a corto y medio plazo es poder implementar este proyecto en un centro educativo real en el que se imparta alguna asignatura de iniciación a la programación. Llevar el proyecto a la práctica me permitiría observar la evolución real del alumnado, recoger datos a través de la evaluación de la situación de aprendizaje y utilizar el feedback de los estudiantes para aplicar mejoras y modificaciones iterativas en el diseño de las sesiones y en las rúbricas de evaluación.

Finalmente, como actuación futura para ampliar este trabajo y consolidar su entidad dentro de una programación anual completa, me plantearía expandir el proyecto hacia la intermodularidad. La IA de ajedrez podría conectarse con otros módulos del grado, por ejemplo, proponiendo a los alumnos que desarrollen una base de datos para almacenar

aperturas en la asignatura de bases de datos, o que diseñen un entorno gráfico de juego en el módulo de diseño de interfaces web. Esto enriquecería el proyecto actual y contribuiría con el uso del ajedrez como herramienta transversal para la enseñanza.

Referencias bibliográficas

- Hita, A. (2025). *Proyecto para crear una IA de ajedrez a nivel formativo entre alumnos del master de desarrollo de aplicaciones web*. <https://github.com/litospayaso/UNIR-ChessIA>
- Hita, A. (2026). *Explicación de como iniciar el proyecto de IA de ajedrez*. <https://youtu.be/DZkaf0yH3aw>
- Bart, W. M. (2014). On the effect of chess training on scholastic achievement. *Frontiers in Psychology*, 5. <https://doi.org/10.3389/fpsyg.2014.00762>
- BOE. (2020). *BOE.es - BOE-A-2020-17264 Ley Orgánica 3/2020, de 29 de diciembre, por la que se modifica la Ley Orgánica 2/2006, de 3 de mayo, de Educación*. <https://www.boe.es/buscar/doc.php?id=BOE-A-2020-17264>
- BOE. (2023a). *I. DISPOSICIONES GENERALES MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL*. <https://www.boe.es/boe/dias/2023/06/03/pdfs/BOE-A-2023-13221.pdf>
- BOE. (2023b). *Real Decreto 405/2023, de 29 de mayo, por el que se actualizan los títulos de la formación profesional del sistema educativo de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma y Técnico Superior en Desarrollo de Aplicaciones Web*. BOE. <https://www.boe.es/boe/dias/2023/06/03/pdfs/BOE-A-2023-13221.pdf>
- BOE. (2024, May 21). *BOE-A-2024-10685 Real Decreto 500/2024, de 21 de mayo, por el que se modifican determinados reales decretos por los que se establecen títulos de Formación Profesional de grado superior y se fijan sus enseñanzas mínimas*. www.boe.es; BOE. <https://www.boe.es/eli/es/rd/2024/05/21/500>
- BOPV. (2011, November 29). *Decreto 245/2011, de 29 de noviembre, por el que se establece el currículo correspondiente al título de Técnico Superior en Desarrollo de*

Aplicaciones Web. Euskadi.eus; BOPV. <https://www.euskadi.eus/web01-bopv/es/bopv2/datos/2011/12/1106111a.pdf>

Colaboradores de los proyectos Wikimedia. (2004, diciembre 21). *Sistema de puntuación Elo*.
https://es.wikipedia.org/wiki/Sistema_de_puntuaci%C3%B3n_Elo

Daysi, C., Silva, C. Z., Maricela, S., & Rubén, M. (2022). Aprendizaje basado en proyectos y la gamificación para generar el aprendizaje activo en los estudiantes. *Revista Ciencia UNEMI*, 15(39), 35–43.
<https://dialnet.unirioja.es/servlet/dcart?info=link&codigo=8439395&orden=0>

Neubauer, A., Ramo Ramiro, M., Roa González, J., & Sánchez Sánchez (Coord.), N. (Eds.). (2023). *Procesos y contextos educativos en educación secundaria*.
<https://dialnet.unirioja.es/servlet/libro?codigo=971659>. Localización: Procesos y contextos educativos en educación secundaria / coord. por Natalia Sánchez Sánchez, 2023, ISBN 9788445445778, págs. 137-155.

Polgar, S. (2018). *WEP2018 TV: How AI turned the chess world upside down*.
<https://www.youtube.com/watch?v=iS9I1IbU15c>

Poot-Delgado, C. A. (2013). RETOS DEL APRENDIZAJE BASADO EN PROBLEMAS.
ResearchGate, 18(2), 307–314.
https://www.researchgate.net/publication/311374934_RETOS_DEL_APRENDIZAJE_BASADO_EN_PROBLEMAS

UNESCO. (2023). *Curriculos de IA para la enseñanza preescolar, primaria y secundaria: un mapeo de los currículos de IA aprobados por los gobiernos*. Unesco.org; UNESCO.
https://unesdoc.unesco.org/ark:/48223/pf0000380602_spa

Anexos