

Videojuego de ajedrez con modo de juego contra la máquina

Anexo III: Estimación del tamaño y esfuerzo

Trabajo de Fin de Grado

GRADO EN INGENIERÍA INFORMÁTICA



**VNiVERSiDAD
D SALAMANCA**

Junio de 2025

Autor

Óscar Sánchez Rubio

Tutores

Luis Augusto Silva Zendron

Gabriel Villarrubia González

Índice

Contenidos

1.	Introducción	1
2.	Modelo de coste	2
3.	Estimación de esfuerzo	3
3.1.	Complejidad de los actores (UAW).....	3
3.2.	Complejidad de los casos de uso (UUCW).....	3
3.3.	Factores de complejidad técnica (TCF)	4
3.4.	Factores de complejidad del entorno (ECF)	6
3.5.	Cálculo de la estimación del esfuerzo.....	8
4.	Bibliografía	11

Ilustraciones

Ilustración 1: Factores de complejidad del entorno	7
Ilustración 2: EZEstimate - Lista de actores y casos de uso	8
Ilustración 3: EZEstimate - Factores de complejidad técnica.....	9
Ilustración 4: EZEstimate - Factores de complejidad del entorno	9
Ilustración 5: EZEstimate - Resultados de la estimación del esfuerzo	10

Tablas

Tabla 1: Complejidad de los actores	3
Tabla 2: Complejidad de los casos de uso	4
Tabla 3: Factores de complejidad técnica	6

1. Introducción

Este anexo tiene como finalidad presentar una estimación del tamaño del sistema a desarrollar, así como del esfuerzo necesario para su implementación. Esta información resulta clave para una adecuada planificación del proyecto, ya que permite prever los recursos humanos, técnicos y temporales requeridos. La estimación se ha realizado a partir de los elementos funcionales definidos durante el análisis del sistema.

2. Modelo de coste

Para estimar el esfuerzo y coste del desarrollo del sistema, se ha optado por utilizar el modelo de coste basado en puntos de casos de uso (*Use Case Points*, UCP) (Moreno García). Este enfoque se trata de una métrica utilizada para evaluar la funcionalidad representada a partir de los casos de uso definidos, considerando tanto la complejidad de las funcionalidades como los actores que interactúan con el sistema.

Este modelo incorpora factores técnicos y ambientales que permiten ajustar la estimación según la naturaleza del proyecto y las características del equipo de desarrollo. A partir del cálculo de los puntos de casos de uso, se podrá derivar una estimación del esfuerzo necesario en meses de persona, y a partir de ahí, estimar el coste total del proyecto.

Para calcular el total de puntos de casos de uso, se requiere de tres variables:

- **UUCP** (*Unadjusted Use Case Points*): Se define como la suma de dos valores.

$$UUCP = UCCW + UAW$$

- **UUCW** (*Unadjusted Use Case Weight*): Evalúa el tamaño del sistema según la cantidad y complejidad de los casos de uso, teniendo en cuenta factores como el número de pasos en los escenarios, la complejidad de la interfaz de usuario, las entidades de base de datos involucradas y las clases implicadas.
- **UAW** (*Unadjusted Actor Weight*): mide el peso de los actores en función de su tipo y nivel de interacción con el sistema; los actores simples son sistemas que se comunican por API, los medios por protocolo (como Internet), y los complejos son usuarios que interactúan mediante una interfaz gráfica.
- **TCF** (*Technical Complexity Factor*): Es un valor que ajusta el tamaño base del sistema en función de su complejidad técnica. Se calcula evaluando 13 aspectos técnicos del proyecto.
- **ECF** (*Environment Complexity Factor*): Ajusta la estimación del esfuerzo según las condiciones del equipo y del entorno de trabajo. Se basa en 8 factores que se puntúan del 0 al 5.

El cálculo de los puntos de casos de uso, los cuales se pueden derivar en los meses persona, se realiza siguiendo la siguiente fórmula:

$$UCP = UUCP * TCF * ECF$$

3. Estimación de esfuerzo

3.1. Complejidad de los actores (UAW)

Para empezar a realizar la estimación temporal, vamos a calcular la complejidad de los distintos actores que van a interactuar con el sistema. Para ello, se recuperarán los distintos actores que interactúan con el sistema, y a cada actor identificado se le asignará una determinada complejidad. Como antes se ha mencionado, esta complejidad puede ser:

- **Simple:** Si el actor es un sistema y la aplicación se comunica con él mediante una API.
- **Medio:** Si el actor es un sistema y la aplicación se comunica con él mediante un protocolo (Internet).
- **Complejo:** Persona con una interfaz gráfica

De los dos actores que hay en este sistema, tan solo uno es complejo, el **actor Usuario**, puesto que, efectivamente, se trata de una persona con una interfaz gráfica. Por otra parte, el **actor Sistema** es un actor simple, puesto que la aplicación no se comunica con él mediante un protocolo.

Actor	Complejidad
Usuario	Complejo
Sistema	Simple

Tabla 1: Complejidad de los actores

3.2. Complejidad de los casos de uso (UUCW)

Para los casos de uso, se trabajará de una forma muy similar a con los actores. Para determinar la complejidad de los casos de uso, se determinará el número de transacciones que tenga cada caso de uso y en función de ese número de **transacciones** se asignará una complejidad a ese caso de uso. Una transacción se define como una o varias acciones realizadas por el usuario seguida de una o varias acciones realizadas por el sistema en respuesta a la acción realizada por el usuario.

Cuando se tiene constancia del número de transacciones de cada caso de uso, se establece su complejidad entre una de estas tres:

- **Simple:** Si un caso de uso tiene menos de 4 transacciones.
- **Medio:** Si un caso de uso tiene entre 4 y 7 transacciones.
- **Complejo:** Si un caso de uso tiene más de 7 transacciones.

Se debe tener en cuenta a las excepciones para realizar el conteo de las transacciones de un caso de uso, puesto que, aunque no pertenecen al flujo normal de pasos del mismo, sí que forman parte de su ejecución normal.

Caso de uso	Transacciones	Complejidad
CU-01 Seleccionar pieza	1	Simple
CU-02 Mostrar movimientos legales para una pieza	1	Simple
CU-03 Deseleccionar pieza	1	Simple
CU-04 Mover pieza	1	Simple
CU-05 Detectar jaque	1	Simple
CU-06 Detectar jaque mate	1	Simple
CU-07 Detectar tablas	1	Simple
CU-08 Finalizar partida	1	Simple
CU-09 Cambiar nombre de jugador	3	Simple
CU-10 Cambiar color de las piezas de los jugadores	1	Simple
CU-11 Elegir modo de juego contra otro jugador	1	Simple
CU-12 Elegir modo de juego contra la máquina	1	Simple
CU-13 Empezar partida	2	Simple
CU-14 Buscar movimientos para la IA	1	Simple
CU-15 Evaluar movimientos	1	Simple
CU-16 Decrementar tiempo restante a los jugadores	1	Simple
CU-17 Elegir dificultad de la IA	2	Simple
CU-18 Cargar distribución del tablero en formato PEN	2	Simple
CU-19 Calcular tiempo de búsqueda	1	Simple

Tabla 2: Complejidad de los casos de uso

3.3. Factores de complejidad técnica (TCF)

En el modelo de puntos de casos de uso, se define 13 factores de complejidad técnica a los que se les debe asignar una complejidad, en función a la complejidad percibida por el equipo de desarrollo. Dicha complejidad será un número entero entre 0 y 5, siendo 0 poco complejo y 5 muy complejo.

Los 13 factores de complejidad técnica son los siguientes:

- **Sistemas distribuidos:** Complejidad respecto a diseñar un sistema distribuido en distintos nodos.
- **Rendimiento:** Complejidad respecto a diseñar un sistema que debe proporcionar las distintas funcionalidades en un tiempo de respuesta óptimo.
- **Eficiencia del usuario final:** Complejidad respecto a diseñar un sistema en el que el usuario debe poder realizar las tareas deseadas con la mayor eficiencia posible.
- **Procesamiento interno complejo:** Complejidad respecto a diseñar un sistema que tiene que realizar una serie de cálculos internos para proporcionar las distintas funcionalidades.
- **Reusabilidad:** Complejidad respecto a reutilizar ciertas porciones de código para aumentar la eficiencia y calidad del sistema, si es que se busca.

- **Facilidad de instalación:** Complejidad respecto a la implementación de una instalación sencilla puede implicar una complejidad técnica adicional.
- **Facilidad de uso:** Complejidad respecto a diseñar un sistema fácil de usar para el usuario.
- **Portabilidad:** Complejidad respecto a diseñar un sistema que funcione en distintas plataformas.
- **Facilidad de cambio:** Complejidad respecto a diseñar un sistema en el que se puedan realizar cambios de manera rápida y sencilla.
- **Concurrencia:** Complejidad respecto a diseñar un sistema que maneja múltiples procesos y/o usuarios de manera concurrente.
- **Características especiales de seguridad:** Complejidad respecto a diseñar un sistema seguro contra posibles ataques.
- **Accesos directo a terceras partes:** Complejidad respecto a diseñar un sistema que permita o proporcione acceso a otros programas ajenos de terceros.
- **Se requiere entrenamiento especial del usuario:** Complejidad respecto a diseñar un sistema que necesite instruir a los usuarios para que estos puedan hacer uso del mismo.

Factor de complejidad técnica	Complejidad	Justificación
Sistemas distribuidos	0	El sistema a desarrollar será una aplicación de escritorio local, por lo que no habrá ningún tipo de comunicación con otros sistemas
Rendimiento	5	El rendimiento es un factor esencial de este sistema, puesto que influye directamente en el desempeño de la inteligencia artificial. Conseguir esto, tampoco es especialmente fácil.
Eficiencia del usuario final	2	La aplicación no requiere que ninguna funcionalidad sea implementada de manera especialmente eficiente para que resulte fácil de manejar para el usuario final. Las interfaces son sencillas e intuitivas
Procesamiento interno complejo	5	Los cálculos correspondientes a la obtención de movimientos legales y a la detección de los jaques o tablas, no son nada triviales
Reusabilidad	2	Aunque no hay grandes partes del código que vayan a ser reusables, es cierto que el tablero aparece en más de una ocasión dentro de distintos ámbitos de la aplicación

Facilidad de instalación	1	La instalación del sistema se basa en descargar el ejecutable, por lo que no es una instalación compleja
Facilidad de uso	2	El sistema es sencillo e intuitivo en su propia definición, no requiere de esfuerzo extra para facilitar su uso
Portabilidad	1	Al desarrollarse en Unity, no se tendrá que hacer especial hincapié en la portabilidad, puesto que este motor gráfico permite que el proyecto sea compatible para Windows, Linux y Mac
Facilidad de cambio	4	Es posible que ciertos aspectos de programa no acaben funcionando y necesiten alterarse, es por ello que cabe prever una facilidad de cambio en estos
Concurrencia	3	A la hora de enfrentarse contra la inteligencia artificial, esta deberá ejecutarse en un hilo secundario, para evitar bloquear la interacción con el usuario del hilo principal de Unity
Características especiales de seguridad	0	El sistema a desarrollar se trata de un motor de ajedrez local, por lo que no hay información relevante que proteger
Accesos directo a terceras partes	0	El sistema no proporcionará servicios a otras plataformas ajenas
Se requiere entrenamiento especial del usuario	0	La aplicación no requerirá de ningún tutorial o guía para su uso, sus funciones están claras

Tabla 3: Factores de complejidad técnica

3.4. Factores de complejidad del entorno (ECF)

Al igual que con los factores de complejidad técnica, se debe asignar un valor de complejidad entre el 0 y el 5 a los 8 factores de complejidad del entorno, los cuáles establecen un medidor de complejidad para la situación en la que se comprende el proyecto.

Los 8 factores de complejidad del entorno son los siguientes:

- **Familiaridad con UML:** Complejidad respecto a la experiencia del equipo de desarrollo con el entorno de UML.
- **Trabajadores a tiempo parcial:** Complejidad respecto a la cantidad de tiempo que pueden dedicar los miembros del equipo al día.
- **Capacidad de los analistas:** Complejidad respecto al conocimiento experto de los analistas.
- **Experiencia en la aplicación:** Complejidad respecto a la experiencia en el desarrollo de aplicaciones similares.

- **Experiencia en orientación a objetos:** Complejidad respecto a la experiencia del equipo de desarrollo en el paradigma de programación orientado a objetos
- **Motivación:** Complejidad respecto a la motivación del equipo en el desarrollo del proyecto.
- **Dificultad del lenguaje de programación:** Complejidad respecto a la dificultad inherente del lenguaje de programación utilizado.
- **Estabilidad de los requisitos:** Complejidad respecto a la estabilidad de los requisitos.

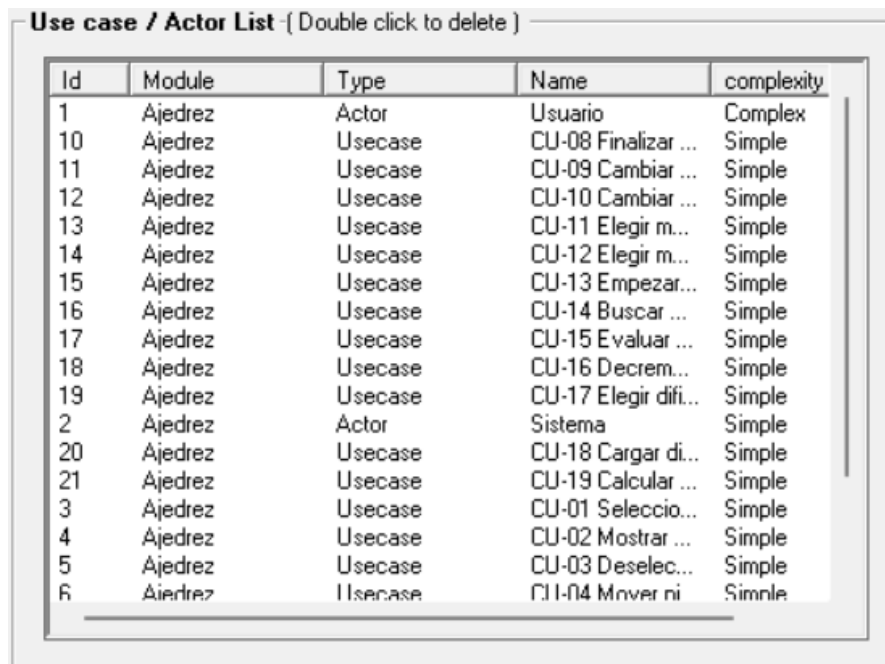
Factor de complejidad del entorno	Complejidad	Justificación
Familiaridad con UML	2	Aunque, durante la carrera de Ingeniería Informática, se han ido cursando varias asignaturas en las que se ha empleado el lenguaje UML, no se considera que el conocimiento en este ámbito sea precisamente elevado
Trabajadores a tiempo parcial	4	Puesto que este trabajo será realizado a la par que el último año de Ingeniería Informática, a dedicación a lo largo del día no podrá ser plena y en exclusiva al mismo
Capacidad de los analistas	4	Debido a la falta de experiencia en el desarrollo de proyectos software a gran escala, la capacidad de análisis puede llegar a ser un problema
Experiencia en la aplicación	2	Gracias a la creación de proyectos personales con la temática del ajedrez, se posee una leve experiencia en trabajar con aplicaciones similares
Experiencia en orientación a objetos	2	De nuevo, a lo largo de la carrera se han empleado diversos lenguajes orientados a objetos, sin embargo, no se considera que la experiencia sea lo suficiente para que la complejidad en este ámbito sea nula
Motivación	1	Dado que la temática del proyecto fue elegida por el propio desarrollador, la motivación para llevar a cabo el proyecto es bastante elevada
Dificultad del lenguaje de programación	1	Intrínsecamente, C# es un lenguaje de alto nivel, que fue diseñado para ser rápido y productivo para los programadores
Estabilidad de los requisitos	4	Debido a que ciertos aspectos de los requisitos pueden no acabar implementándose de forma exitosa en la versión final de la aplicación, es posible que algunos de ellos necesiten ser alterados

Ilustración 1: Factores de complejidad del entorno

3.5.Cálculo de la estimación del esfuerzo

Una vez se han obtenido los valores de todas las variables, para llevar a cabo el cálculo de la estimación del esfuerzo y obtener los puntos de casos de uso del proyecto, se va a hacer uso de la herramienta **EZEstimate** (EZEstimate), la cual, permite estimar el esfuerzo de un proyecto a partir de dichas variables.

En primer lugar, se introducirán los datos relativos a los actores y los casos de uso, indicando, para cada uno de ellos, su nombre y su complejidad asignada. La lista de actores y casos de uso de EZEstimate es la siguiente:



Id	Module	Type	Name	complexity
1	Ajedrez	Actor	Usuario	Complex
10	Ajedrez	Usecase	CU-08 Finalizar ...	Simple
11	Ajedrez	Usecase	CU-09 Cambiar ...	Simple
12	Ajedrez	Usecase	CU-10 Cambiar ...	Simple
13	Ajedrez	Usecase	CU-11 Elegir m...	Simple
14	Ajedrez	Usecase	CU-12 Elegir m...	Simple
15	Ajedrez	Usecase	CU-13 Empezar...	Simple
16	Ajedrez	Usecase	CU-14 Buscar ...	Simple
17	Ajedrez	Usecase	CU-15 Evaluar ...	Simple
18	Ajedrez	Usecase	CU-16 Decrem...	Simple
19	Ajedrez	Usecase	CU-17 Elegir difi...	Simple
2	Ajedrez	Actor	Sistema	Simple
20	Ajedrez	Usecase	CU-18 Cargar di...	Simple
21	Ajedrez	Usecase	CU-19 Calcular ...	Simple
3	Ajedrez	Usecase	CU-01 Seleccio...	Simple
4	Ajedrez	Usecase	CU-02 Mostrar ...	Simple
5	Ajedrez	Usecase	CU-03 Deselec...	Simple
6	Ajedrez	Usecase	CU-04 Mover pi...	Simple

Ilustración 2: EZEstimate - Lista de actores y casos de uso

En segundo lugar, se introducirán los datos relativos a los factores de complejidad técnica, indicando, para cada uno de ellos, su complejidad percibida. La lista de factores de complejidad técnica de EZEstimate es la siguiente:

Technical complexity factors	
Factor	Relevance
Distributed system	<input type="text" value="0"/>
Response / Throughput performance objectives	<input type="text" value="5"/>
End-user efficiency	<input type="text" value="2"/>
Complex internal processing	<input type="text" value="5"/>
Reusable code	<input type="text" value="2"/>
Easy to install	<input type="text" value="1"/>
Easy to use	<input type="text" value="2"/>
Portable	<input type="text" value="1"/>
Easy to change	<input type="text" value="4"/>
Concurrent	<input type="text" value="3"/>
Includes security features	<input type="text" value="0"/>
Third party access	<input type="text" value="0"/>
Special user training facilities required	<input type="text" value="0"/>

Ilustración 3: EZEstimate - Factores de complejidad técnica

En tercer lugar, se introducirán los datos relativos a los factores de complejidad del entorno, indicando, para cada uno de ellos, su complejidad percibida. La lista de factores de complejidad del entorno de EZEstimate es la siguiente:

Environmental factors	
Factor	Relevance
Familiar with Rational unified process	<input type="text" value="2"/>
Application experience	<input type="text" value="2"/>
Object oriented experience	<input type="text" value="2"/>
Lead analyst capability	<input type="text" value="4"/>
Motivation	<input type="text" value="1"/>
Stable requirements	<input type="text" value="4"/>
Part-time workers	<input type="text" value="4"/>
Difficult programming language	<input type="text" value="1"/>

Ilustración 4: EZEstimate - Factores de complejidad del entorno

Con todos estos datos, EZEstimate calculará el total de puntos de casos de uso, siguiendo la fórmula que apareció al inicio de este anexo. Suponiendo, por defecto, un total de 20 horas de persona por punto de caso, el resultado mostrado por la herramienta es el siguiente:

Estimation Summary	
UAW	4
UUCW	95
UUPC = UAW + UUCW	99
TFactor	30
EFactor	12
TCF = $0.6 + (.01 * \text{TFactor})$	0.9
EF = $1.4 + (-0.03 * \text{EFactor})$	1.04
UCP = UUCP * TCF * EF	92,664
Total Effort@ 20 Hrs/UCP	1853,28

Ilustración 5: EZEstimate - Resultados de la estimación del esfuerzo

El resultado de la estimación del esfuerzo es un total de 1853,28 horas de persona. Teniendo en cuenta que un mes de persona son aproximadamente 150 horas de persona, el proyecto constaría en total 12,36 meses de persona.

4. Bibliografía

EZEstimate. (n.d.). Retrieved from

<https://studium24.usal.es/mod/resource/view.php?id=67601>

Moreno García, M. N. (n.d.). Transparencia de Gestión de Proyectos, Práctica 1 - Estimación del esfuerzo. Salamanca. Retrieved from

https://studium24.usal.es/pluginfile.php/96118/mod_resource/content/1/GP-GII_Primer_practica.pdf