

# Historias de Usuario

<b>Tablero</b>	<b>3</b>
<b>Crear fichas</b>	<b>4</b>
<b>Elegir blancas y negras</b>	<b>5</b>
<b>Mover pieza (básico)</b>	<b>6</b>
<b>Posición de piezas iniciales</b>	<b>7</b>
<b>Comer</b>	<b>8</b>
<b>Jaque</b>	<b>9</b>
<b>Jaque mate</b>	<b>10</b>
<b>Movimiento inicial de peones</b>	<b>11</b>
<b>Posibles movimientos</b>	<b>12</b>
<b>Enroque</b>	<b>13</b>
<b>Coronar</b>	<b>14</b>
<b>Comer al paso</b>	<b>15</b>
<b>Tablas</b>	<b>16</b>
<b>Pedir y aceptar tablas</b>	<b>17</b>
<b>Rendirse</b>	<b>18</b>
<b>Mostrar reglas básicas</b>	<b>19</b>
<b>Banquillo</b>	<b>19</b>
<b>Historial de jugadas</b>	<b>21</b>
<b>Colorear casillas posibles</b>	<b>22</b>
<b>Mensajes de error</b>	<b>23</b>
<b>Toolbar</b>	<b>23</b>
<b>Vista de turno</b>	<b>24</b>
<b>Salto del peón</b>	<b>26</b>
<b>Contador</b>	<b>26</b>
<b>Muerte rey</b>	<b>27</b>

<b>Jugar online</b>	<b>29</b>
<b>Programación de la IA</b>	<b>30</b>
<b>Interfaz elegir modo</b>	<b>31</b>
<b>Reiniciar partida</b>	<b>31</b>
<b>Casillas amenazadas</b>	<b>32</b>
<b>Salir del juego</b>	<b>33</b>

## 1. Tablero

**Usuario:**

Cliente

**Descripción:**

Como usuario quiero un tablero para poder en el mismo realizar el movimiento de las fichas y tener un apoyo gráfico a la hora de jugar.

**Estimación:** 1 semana

**Prioridad:** Alta

**Riesgo en desarrollo:** Baja

**Programador responsable:** -----

**Criterios de aceptación:**

- El tablero debe ser de 8x8.
- Debe estar debidamente numerado (de la A a la H en horizontal y del 1 al 8 en vertical).
- El tablero debe iniciarse al principio del programa y mantenerse invariable hasta que se finalice el mismo.
- Debe haber una matriz asociada al tablero que se actualizará según la posición de las piezas en cada turno y se limpiará e inicializará al comenzar una partida.

## 2. Crear fichas

**Usuario:**

Programador

**Descripción:**

Como programador quiero crear las diferentes fichas del ajedrez para poder más adelante añadirles sus respectivas funciones a la hora de jugar una partida.

**Estimación:** 1 semanas

**Prioridad:** Alta

**Riesgo en desarrollo:** Bajo

**Programador responsable:** ----

**Criterios de aceptación:**

- Deben tener una variable que las defina como blancas o negras
- Deben tener su movimiento propio
- Deben tener su posicionamiento inicial
- Deben tener su rol/ nombre de pieza definido
- Debe tener una posición actual

### **3. Elegir blancas y negras**

**Usuario:**

Cliente

**Descripción:**

Como cliente quiero poder elegir el color que jugaré en la partida. En caso de no llegar a un acuerdo con el rival, el programa lo elegirá aleatoriamente.

**Estimación:** 1 semana

**Prioridad:** Alta

**Riesgo en desarrollo:** Bajo

**Programador responsable:** ----

**Criterios de aceptación:**

- Los dos jugadores han de estar de acuerdo en quién empieza. En caso de no estarlo, se decide aleatoriamente.

## 4. Mover pieza (básico)

**Usuario:**

Cliente

**Descripción:**

Como cliente quiero ser capaz de hacer los diferentes movimientos de cada pieza en el tablero para poder realizar los movimientos básicos de las fichas en un tablero vacío.

**Estimación:** 2 semanas

**Prioridad:** Alta

**Riesgo en desarrollo:** Bajo

**Programador responsable:** ----

**Criterios de aceptación:**

- Se debe describir el movimiento básico de cada ficha
- Se debe facilitar la futura implementación de movimientos más complejos
- Se deben tener en cuenta los bordes del tablero
- Se debe actualizar la posición de la ficha tanto para la misma como en la matriz del tablero

## 5. Posición de piezas iniciales

**Usuario:** Cliente

**Descripción:**

Como cliente quiero tener al iniciar una partida las piezas colocadas en sus respectivas posiciones iniciales para poder empezar una nueva partida si así lo deseo.

**Estimación:** 1 semana

**Prioridad:** Alta

**Riesgo en desarrollo:** Bajo

**Programador responsable:** ----

**Criterios de aceptación:**

- Las piezas blancas deben estar en el lado de las blancas (filas 1 y 2) y las negras en el de las negras (filas 7 y 8)
- Cada pieza debe ocupar su respectivo sitio tal y como dictan las reglas de ajedrez
- Se debe actualizar la posición de las piezas tanto en sí mismas como en la matriz del tablero
- Deben ser el número de fichas iniciales correcto por cada color (8 peones, 2 torres, 2 caballos, 2 alfiles, la reina y el rey)

## **6. Comer**

**Usuario:** Cliente

**Descripción:**

Como usuario poder eliminar una pieza enemiga al mover una pieza mía en su misma posición para poder eliminar piezas del contrincante y mover mi pieza.

**Estimación:** 2 semanas

**Prioridad:** Alta

**Riesgo en desarrollo:** Medio

**Programador responsable:** ----

**Criterios de aceptación:**

- Una pieza solo puede comer a una pieza de otro color.
- Al comer, se debe llevar un registro banquillo de las pieza que se ha comido.
- Se debe actualizar la posición de la pieza que ha comido, ocupando la casilla de la pieza comida y dejar libre la casilla que ocupaba.



## 7. Jaque

**Usuario:**

Cliente

**Descripción:**

Como usuario quiero saber cuando mi rival amenaza a mi rey y puedo realizar algún movimiento este mismo turno para romper la amenaza para evitar que haga jaque mate y perder la partida.

**Estimación:** 2 semanas

**Prioridad:** Alta

**Riesgo en desarrollo:** Alto

**Programador responsable:** ----

**Criterios de aceptación:**

- Debe haber alguna posibilidad de que el rey pueda evitar ser comido realizando alguna jugada.
- El jugador debe evitar que el rey sea comido, es decir, solo puede hacer una jugada que evite dicho suceso.
- Deben tenerse en cuenta las posiciones de las demás fichas enemigas para que al mover el rey no pueda ser comido por otra de las fichas enemigas.

## 8. Jaque mate

**Usuario:** Cliente

**Descripción:**

Como usuario quiero saber cuando uno de los dos jugadores no puede hacer más movimientos, estando previamente amenazado, sin que su rey no esté expuesto a ser comido para saber qué jugador ha ganado la partida.

**Estimación:** 2 semanas

**Prioridad:** Alta

**Riesgo en desarrollo:** Alto

**Programador responsable:** ----

**Criterios de aceptación:**

- Al hacer jaque mate, automáticamente sale el ganador de la partida impidiendo mover más piezas del tablero.
- No hay posibilidad de que el rey previamente amenazado se mueva y no sea comido.

## 9. Movimiento inicial de peones

**Usuario:** Cliente

**Descripción:**

Como usuario quiero poder elegir si muevo una o dos casillas de un peón seleccionado siempre que siga en su casilla inicial.

**Estimación:** 1 semana

**Prioridad:** Media

**Riesgo en desarrollo:** Bajo

**Programador responsable:** ----

**Criterios de aceptación:**

- No debe haber ninguna pieza que obstaculice el movimiento del peón.
- Debe estar el peón en la casilla inicial para poder realizar el movimiento doble.
- Debe actualizarse la casilla de la pieza una vez se haya movido.

## 10. Posibles movimientos

**Usuario:** Programador

**Descripción:**

Como programador quiero una crear una lista con todos los posibles movimientos de cada pieza.

**Estimación:** 1 semana

**Prioridad:** Alta

**Riesgo en desarrollo:** Medio

**Programador responsable:** ----

**Criterios de aceptación:**

- Se debe tener en cuenta el tipo de movimiento dependiendo de la pieza seleccionada.
- Se debe tener en cuenta las piezas que hay repartidas por el tablero.

## 11. Enroque

**Usuario:** Cliente

**Descripción:**

Como usuario quiero realizar el movimiento “especial” del enroque para poder cubrir una mayor gama de movimientos y defender al rey.

**Estimación:** 1 semana

**Prioridad:** Media

**Riesgo en desarrollo:** Medio

**Programador responsable:** Silvia y Javi

**Criterios de aceptación:**

- Ni el rey ni la torre objetivo deben haberse movido previamente
- No debe haber piezas entre medias del rey y la torre
- El rey no puede estar amenazado para poder realizar el movimiento
- Ninguna casilla entre medias debe estar amenazada por una pieza rival para poder completar el movimiento
- Se debe actualizar correctamente la posición de las piezas

## 12. Coronar

**Usuario:**

Cliente

**Descripción:**

Como cliente, quiero que una vez un peón avance hasta el final del tablero, pueda transformarse en otra pieza que yo decida (alfil, reina, caballo o torre) para poder tener una pieza mejor en el tablero que me facilite la victoria.

**Estimación:** 1 semana

**Prioridad:** Media

**Riesgo en desarrollo:** Medio

**Programador responsable:** Pablo

**Criterios de aceptación:**

- El peón debe estar en la última fila del tablero.
- Se debe seleccionar a la pieza que se quiere coronar.
- Se debe eliminar el peón coronado y añadir la nueva pieza.

## 13. Comer al paso

**Usuario:**

Cliente

**Descripción:**

Como cliente quiero que cuando un peón avance dos casillas iniciales, otro peón enemigo que esté adyacente en la misma fila pueda comerlo moviéndose a la posición anterior al primer peón para poder eliminar esta pieza.

**Estimación:** 1 semanas

**Prioridad:** Baja

**Riesgo en desarrollo:** Alto

**Programador responsable:** ----

**Criterios de aceptación:**

- El peón debe haber realizado su movimiento inicial de mover dos casillas.
- Debe haber un peón enemigo adyacente al peón movido en su misma fila para poder completar este movimiento.
- El movimiento al comer debe ser exclusivamente a la posición anterior del peón que se va a eliminar.

## 14. Tablas

**Usuario:**

Programador

**Descripción:**

Como programador quiero hacer que el programa finalice la partida al detectar un empate para que en caso de no poder terminar la partida de forma natural, se finalice el programa evitando una partida infinita.

**Estimación:** 4 semanas

**Prioridad:** Media

**Riesgo en desarrollo:** Alta

**Programador responsable:** ----

**Criterios de aceptación:**

- Ninguno de los jugadores ha de ser capaz de ganar.
- Normas estándar de las tablas del ajedrez.



## **15. Pedir y aceptar tablas**

**Usuario:** Cliente

**Descripción:**

Como usuario, quiero poder pedir tablas a mi rival para terminar una partida en empate si el rival acepta las mismas.

**Estimación:** 1 semana

**Prioridad:** Baja

**Riesgo en desarrollo:** Bajo

**Programador responsable:** ----

**Criterios de aceptación:**

- El rival deberá de aceptar la petición de tablas.
- No se puede pedir tablas cuando la persona que lo va a pedir se encuentra en jaque mate.
- Solo se podrá pedir tablas en el turno del jugador.

## 16. Rendirse

**Usuario:** Cliente

**Descripción:**

Como usuario, quiero poder rendirme para terminar la partida en cualquier momento, dándole la victoria al rival.

**Estimación:** 1 semana

**Prioridad:** Baja

**Riesgo en desarrollo:** Bajo

**Programador responsable:** Carlos

**Criterios de aceptación:**

- Debe ser el turno del jugador que desea rendirse.
- Automáticamente el rival ganará la partida y no se podrán mover más piezas del tablero

## **17. Mostrar reglas básicas**

**Usuario:** Cliente

**Descripción:**

Como usuario, quiero poder consultar las reglas básicas del ajedrez (movimiento de piezas, normas...) para poder informarme en mi turno de las mismas.

**Estimación:** 1 semana

**Prioridad:** Baja

**Riesgo en desarrollo:** Bajo

**Programador responsable:** -----

**Criterios de aceptación:**

- Debe ser el turno del jugador que desea ver las reglas.

## 18. Banquillo

**Usuario:** Cliente

**Descripción:**

Como usuario, quiero poder ver las piezas eliminadas a lo largo de la partida para revisar el estado del juego.

**Estimación:** 1 semana

**Prioridad:** Baja

**Riesgo en desarrollo:** Bajo

**Programador responsable:** Belén

**Criterios de aceptación:**

- Se podrá ver a lo largo de toda la partida
- Se actualizará a medida que se vayan eliminando las piezas.

## 19. Historial de jugadas

**Usuario:** Cliente

**Descripción:**

Como usuario quiero poder acceder a un historial de jugadas para poder ver reflejadas los movimientos previos de la partida

**Estimación:** 1 semana

**Prioridad:** Baja

**Riesgo en desarrollo:** Bajo

**Programador responsable:** Miguel

**Criterios de aceptación:**

- Deben guardarse las jugadas con el formato adecuado después de cada turno
- Debe poderse mostrar en cualquier momento de la partida

## **20. Colorear casillas posibles**

**Usuario:** Cliente

**Descripción:**

Como cliente quiero poder ver las casillas disponibles a las que se pueda mover la pieza seleccionada.

**Estimación:** 1 semana

**Prioridad:** Media

**Riesgo en desarrollo:** Bajo

**Programador responsable:** Pablo y Javi

**Criterios de aceptación:**

- Debe haber movimientos posibles de la figura.

## 21. Mensajes de error

**Usuario:** Programador

**Descripción:**

Como programador, quiero que cada vez que se intente un movimiento imposible salte un mensaje de error que informe al jugador.

**Estimación:** 1 semana

**Prioridad:** Medio

**Riesgo en desarrollo:** Bajo

**Programador responsable:** Miguel

**Criterios de aceptación:**

- Deben estar creadas todas las excepciones.
- Los mensajes deben salir cada vez que se intente hacer un movimiento imposible.

## **22. Toolbar**

**Usuario:** Cliente

**Descripción:**

Como cliente quiero que haya una parte de la pantalla en la que haya opciones como salir del juego, resetear o pedir tablas.

**Estimación:** 1 semana

**Prioridad:** Medio

**Riesgo en desarrollo:** Bajo

**Programador responsable:** Belén

**Criterios de aceptación:**

- La toolbar debe contener botones que permitan la posibilidad de salir del juego, resetear o pedir tablas en el turno de cada jugador.
- Deben estar implementados dichos métodos.



## **23. Vista de turno**

**Usuario:** Cliente

**Descripción:**

Como cliente quiero poder ver en todo momento de quién es el turno.

**Estimación:** Una semana

**Prioridad:** Media

**Riesgo en desarrollo:** Bajo

**Programador responsable:** Belén

**Criterios de aceptación:**

- Debe haberse implementado el cambio de turno (mediante un contador).

## 24. Salto del peón

**Usuario:** Programador

**Descripción:**

Como programador quiero hacer que los peones se puedan mover dos casillas hacia delante en su primer movimiento si el usuario lo desea.

**Estimación:** 1 semana

**Prioridad:** Media

**Riesgo en desarrollo:** Bajo

**Programador responsable:** Javi y Silvia

**Criterios de aceptación:**

- El peón debe estar en su casilla inicial sin haber realizado ningún movimiento.
- Las dos casillas a las que se puede mover el peón deben estar vacías
- Una vez movido el peón, la posición del mismo se actualiza

## **25. Contador**

**Usuario:** Programador

**Descripción:**

Como programador quiero hacer que aparezca un contador ascendente de turnos, que indique cuántos llevan.

**Estimación:** 1 semana

**Prioridad:** Baja

**Riesgo en desarrollo:** Muy bajo

**Programador responsable:** Carlos y Jorge

**Criterios de aceptación:**

- Debe empezarse una partida.

## **26. Muerte rey**

**Usuario:** Cliente

**Descripción:**

Como cliente, al comer al rey quiero que se me dé la posibilidad de empezar una partida nueva o dejar de jugar

**Estimación:** 1 semana

**Prioridad:** Alta

**Riesgo en desarrollo:** Bajo

**Programador responsable:** Carlos

**Criterios de aceptación:**

- Tiene que haber un jaque mate previo.

## **27. Jugar online**

**Usuario:** Cliente

**Descripción:**

Como cliente, quiero tener la posibilidad de jugar online con otra persona.

**Estimación:** 2 semanas

**Prioridad:** Alta

**Riesgo en desarrollo:** Alto

**Programador responsable:** Miguel, Belen, Pablo

**Criterios de aceptación:**

- El juego me debe ofrecer la posibilidad de conectarme a la red.

## 28. Programación de la IA

**Usuario:** Programador

**Descripción:**

Como programador quiero crear una IA que actúe como jugador rival, mediante una mezcla de movimientos aleatorios y jugadas previamente programadas.

**Estimación:** 2 semanas

**Prioridad:** Alta

**Riesgo en desarrollo:** Alto

**Programador responsable:** Javi y Pablo

**Criterios de aceptación:**

- Se deben planear con antelación las jugadas que va a realizar la IA.
- Se debe crear un random de movimientos.

## **29. Interfaz elegir modo**

**Usuario:** Cliente

**Descripción:**

Como jugador quiero poder escoger el modo de juego.

**Estimación:** 1 semana

**Prioridad:** Alta

**Riesgo en desarrollo:** Bajo

**Programador responsable:** Pablo

**Criterios de aceptación:**

- Para su correcto funcionamiento, se recomienda la implementación previa de los diferentes modos de juego (online, juego contra una IA o contra otro jugador).

### **30. Reiniciar partida**

**Usuario:** Programador

**Descripción:**

Como programador quiero hacer que se pueda reiniciar la partida en el momento deseado.

**Estimación:** 1 semana

**Prioridad:** Media

**Riesgo en desarrollo:** Medio

**Programador responsable:** Carlos, Pablo

**Criterios de aceptación:**

- Debe haber una partida iniciada.
- Se debe permitir solo para el turno del jugador actual.



## **31. Casillas amenazadas**

**Usuario:** Programador

**Descripción:**

Como programador quiero hacer que se puedan ver todas aquellas casillas amenazadas por el rival.

**Estimación:** 1 semana

**Prioridad:** Alta

**Riesgo en desarrollo:** Medio

**Programador responsable:** Javi y Miguel

**Criterios de aceptación:**

- Deben estar completado el coloreado de casillas para poder indicarlás.

## **32. Salir del juego**

**Usuario:** Cliente

**Descripción:**

Como cliente quiero ser capaz de salir del juego en cualquier momento, independientemente del estado de la partida.

**Estimación:** 1 semana

**Prioridad:** Media

**Riesgo en desarrollo:** Bajo

**Programador responsable:** ----

**Criterios de aceptación:**

No constan.