



PROYECTO INGENIERIA DE SOFTWARE

TRICOTECH



Asignatura: Ingeniería de software

Nombre del docente: Andrés Aranda

Nombre estudiantes: Jose Madruga, Fabricio Vanrell y Facundo benitez

Curso: 3ºMG

Institución educativa: ITI

Fecha de entrega: 15/09/25



ÍNDICE

1. Relevamiento.....	3
1.1 Formulario relevamiento.....	3
1.2 Técnicas utilizadas.....	3
2. Estudio de factibilidades.....	3
2.1 Factibilidad Operativa.....	4
2.2 Factibilidad Técnica	4
2.3 Factibilidad Económica	5
2.4 Factibilidad Legal	5
3. Paradigma de conformación del equipo.....	6
4. Especificación de Requerimientos	6
5. Implementación del modelo de desarrollo.....	6
6. Lógica del sistema.....	7
7. Documentación de inicio y planificación de proyecto.....	8
7.1 Planificación inicial.....	8
8. Paradigma de conformación en coordinación con UTULab.....	8
9. Diagrama de Gantt.....	9
10. Diagrama de PERT.....	10
11. Prototipado de la aplicación.....	11
12. Carta de presentación del equipo.....	11
13. Reglamento Interno del equipo.....	11
14. Creación repositorio Git.....	11
15. Tabla de decisiones.....	12
16. Diagrama de Gantt 2ºEntrega.....	13
17. Diagrama casos de uso.....	14
18. Plantilla de casos de uso.....	15-18
19. Diagrama de estados.....	18-19
20. Estimaciones.....	19-20



1. Relevamiento

Para el revelamiento es muy importante saber que le puedo preguntar al cliente y que necesidades este tiene, mediante la entrevista, la observación, en este caso la propuesta del proyecto ya está estructurada y entregada por la UTU con sus respectivos requerimientos. Si lo que necesitamos no estuviera en la letra del proyecto le preguntaría al cliente:

1.1 Formulario relevamiento

- ¿Hay un reglamento base ya definido del juego o lo tenemos que desarrollar desde cero?
- ¿Deseas un modo de juego de seguimiento?
- ¿Cuáles son los objetivos del juego? ¿Cómo se gana?
- ¿Qué mecánicas tiene que tener el juego, por Ejemplo: draft, colocación de fichas, puntuación por zonas...
- ¿Cuántos jugadores pueden jugar a la vez el juego?
- ¿Cuál será la duración promedio de una partida?
- ¿El juego es una creación original o una adaptación del Draftosaurus ya existente?

1.2 Técnicas utilizadas

Utilizamos las técnicas de observación, principalmente la de observar atentamente los casos, tomamos información y la registramos para su posterior análisis. En nuestro caso la observación es participante porque nos involucramos y formamos parte del proyecto.

2. Estudio de factibilidades

Para hacer un estudio de factibilidades lo primero que debemos determinar es la factibilidad operativa, técnica, económica y legal del proyecto. Preguntarnos ¿Es factible desarrollar este proyecto de Draftosaurus operativamente para nosotros? ¿Es Útil? ¿Contamos con los conocimientos necesarios?

2.1 Factibilidad Operativa

La aplicación será intuitiva y accesible, con una interfaz sencilla que no requiere conocimientos técnicos avanzados. Esto garantiza que quienes la usen puedan adaptarse rápidamente a su funcionamiento sin necesidad de una capacitación extensa. En cuanto al el mantenimiento y soporte pueden ser



garantizados por el equipo de desarrollo, que cuenta con los conocimientos necesarios en PHP, HTML, CSS y JavaScript para realizar actualizaciones y correcciones. Esto asegura la continuidad operativa del sistema.

Nuestra conclusión: La factibilidad operativa del proyecto es alta, dado que existe aceptación por parte de los usuarios, el sistema resulta sencillo de utilizar, se integra adecuadamente con los procesos actuales y puede ser mantenido por el equipo desarrollador sin necesidad de recursos externos adicionales.

2.2 Factibilidad Técnica

En este caso es un proyecto web por lo que no necesitaremos crear una app y teniendo en cuenta eso tampoco necesitaremos internet para jugarlo local. El juego es tanto para computadora, celular y Tablet. Para que el juego sea jugable en dichos dispositivos lo vamos a hacer responsive lo que lo hace adaptable a cualquier tipo y tamaño de pantallas.

Software mínimo para realizar el juego en la siguiente tabla:

Tipo de software	Recomendado	Función principal
Sistemas Operativos	Windows 7 en adelante, macOS, Linux, Android, iOS	Entorno de desarrollo
Editor de Código	Visual Studio Code o similar	Escribir y editar HTML, CSS, JS
Servidor Local	XAMPP 8.X o servidor escolar	Ejecutar código PHP localmente (simula un servidor web)
Navegador Web	Chrome 60+, Mozilla Firefox 55+, Safari 10+, Brave y Edge actualizados 2017 mínimo. Prácticamente todos menos Explorer ya que está obsoleto y no soporta características modernas	Probar como se ve y funciona el juego
Control de Versiones	Git + GitHub	Guardar avances del proyecto y colaborar en equipo
Diseño gráfico	GIMP, Canva	Crear o editar fichas, tableros, fondos, etc.



Software mínimo para poder jugar:

Software mínimo para poder jugar es exactamente lo mismo que necesitamos nosotros los desarrolladores, pero omitiendo lo siguiente: Visual Studio Code, GitHub o parte del diseño gráfico como GIMP o Canva.

Hardware mínimo para poder desarrollar / jugar el juego en la siguiente tabla:

Componente	Mínimo requerido (funciona)	Recomendado
Procesador (CPU)	Intel Pentium 4 / AMD equivalente (1.5 GHz)	Intel i3 o superior (2 GHz o más)
Memoria RAM	2 GB	4 GB o más
Almacenamiento	500 MB libres (para sistema + navegador)	2 GB libres (si se instala servidor local)
Pantalla	Resolución 1024x768	1366x768 o superior
Conexión a internet	No es obligatoria si es local	Para documentación, librerías o jugarlo online
Placa de video	Grafica integrada estándar	Cualquiera por encima de lo estándar está muy bien porque el juego no está en 3D con ultra gráficos

2.3 Factibilidad Económica

Este proyecto se va a llevar a cabo sin fines de lucro y tampoco tendríamos la obligación de gastar dinero en impresiones, prototipos o materiales ya que la institución nos brinda lo necesario.

2.4 Factibilidad Legal

Al ser un juego ya creado con su reglamento y marca registrada nosotros no vamos a utilizar este juego con fines de lucro como ya mencionamos en la anterior factibilidad ni para sustituir a la obra original, se desarrollará una versión propia, con diseño original, adaptada al contexto educativo



3. Paradigma de conformación del equipo

Este equipo de 3 personas está organizado por el Paradigma Descentralizado democrático (DD) porque no tenemos un líder permanente en si que tome solo las decisiones, se nombran coordinadores para tareas a corto plazo y nos sustituimos entre nosotros para diferentes tareas. Las decisiones finales se toman en consenso, seríamos un paradigma como la siguiente imagen:



4. Especificación de Requerimientos

https://drive.google.com/file/d/1i8bVbjJcAsjNzkt1yiA0S8lULc1pjA5a/view?usp=drive_li nk

5. Implementación del modelo de desarrollo

Para este proyecto se eligió el modelo de desarrollo iterativo ya que nos permite ir construyendo el sistema por partes y mejorarlo poco a poco hasta la entrega final. La idea es dividir el sistema en módulos (por ejemplo: registro de jugadores, gestión de partidas, tablero, puntajes) y en cada iteración desarrollar una parte, probarla y mejorarla antes de seguir con la siguiente. Así siempre vamos teniendo una versión funcional, aunque no esté completa.

Ventajas de este modelo:

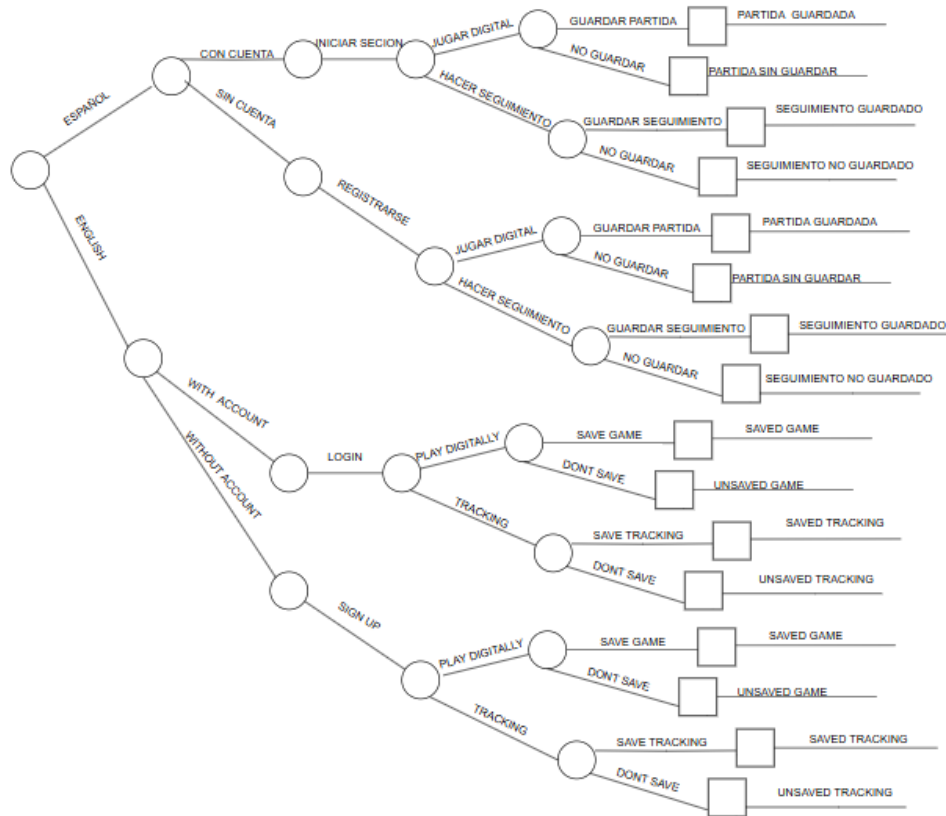
- Podemos mejorar lo ya hecho sin necesidad de esperar al final.
- Permite ver avances concretos en cada etapa.
- Reduce errores porque se va probando de a poco.
- Es flexible si aparece algún cambio en los requisitos.

En conclusión, este modelo se adapta mejor a nuestro proyecto porque trabajamos paso a paso, haciendo entregas parciales y corrigiendo lo necesario hasta lograr la versión final.



6. Lógica del sistema

Árbol de decisión:



El siguiente árbol esta generado a partir de las siguientes decisiones:

- Selección de idioma
- Si ya tiene cuenta o tiene que registrarse
- Va a jugar digital o va a hacer seguimiento?
- Guardar partida o no

Mas adelante agregaremos más decisiones, pero en un principio seria como esta detallado en la imagen



7. Documentación de inicio y planificación del proyecto

7.1 Planificación inicial

Nuestra planificación inicial fue leer bien la letra cada uno de los integrantes del grupo para entender bien lo que había que hacer, para luego hacer una división de materias por cada uno de nosotros, el plan era cada uno de los integrantes tiene que hacer los requerimientos de 2 materias de la rama tecnológica con las que nos sintiéramos más cómodos y las materias de la rama común las haríamos en conjunto vía Discord o en persona. En el grupo de Discord y WhatsApp se sube material y las tareas realizadas. Nos planteamos objetivos a corto y largo plazo para las diferentes entregas. En las llamadas de Discord hacemos los requerimientos de las materias de la rama común.

Cuando ya tenemos los requerimientos completados de nuestras materias asignadas hacemos llamada en Discord y cada uno de los integrantes muestra su trabajo, explica como hizo en cada uno de los requerimientos y responde dudas de los compañeros. Una vez ya terminada la explicación se recopila toda la documentación para volver a leerla y que este la completa aceptación de todos los integrantes y así lograr las entregas en tiempo y forma.

8. Paradigma de conformación del equipo (Coordinación con UTULab)

Como no hablamos aun de ningún paradigma en UTULab subo el mismo que para Ingeniería de Software. Este equipo está organizado por el conjunto entre el Paradigma de Programador jefe y el Paradigma Abierto. Elegimos Programador jefe porque en el equipo hay un líder que es la autoridad y es el Coordinador del equipo, también tenemos al ayudante del líder que en nuestro caso es el Subcoordinador del equipo y el resto del grupo que son dos integrantes que son quienes llevan a cabo las tareas delegadas por el Coordinador y/o Subcoordinador. Y el modelo de Paradigma Abierto porque en el equipo hay tareas que se llevan a cabo grupalmente en base a mucha comunicación y la toma de decisiones es consensuada.



9. Diagrama de Gantt primera entrega

La entrega inicio el 17/06/2025 finaliza el día 07/07/2025

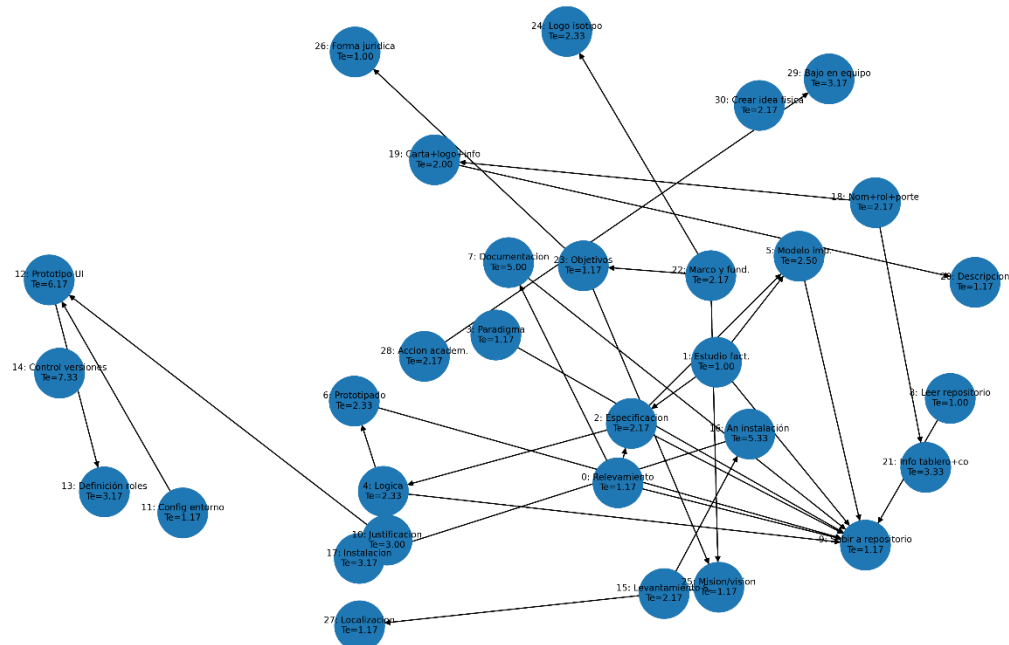
(va adjunto el archivo del diagrama en GanntProject)





10. Diagrama de PERT con GanttProject

Diagrama PERT (duración total estimada ≈ 12.33 unidades de tiempo)



$$te = o + 4m + b / 6 \quad \sigma^2 = (b-o)/6)^2$$

	A	B	C	D	E	F	G	H
1	Actividad	Predecesor	ID	o	m	b	Te	O ²
2	Relevamiento	x	0	0	1	3	1.16	0.25
3	Estudio fact.	x	1	0	1	2	1	0.1
4	specificación	0 y 1	2	1	2	4	2.16	0.25
5	Paradigma	x	3	0	1	3	1.16	0.25
6	Lógica	2	4	1	2	5	2.3	0.4
7	Modelo imp.	1 y 2	5	1	2	6	2.5	0.6
8	Prototipado	4	6	1	2	5	2.3	0.4
9	documentación	0	7	2	5	8	5	1
10	crear repositorio	x	8	0	1	2	1	0.1
11	crear a repositorio	TODAS	9	0	1	3	1.6	0.25
12	Justificación	x	10	1	3	5	3	0.4
13	configuración entorno	x	11	0	1	3	1.16	0.25
14	Prototipo UI	10 y 11	12	4	6	9	6.16	0.6
15	levantamiento	12	13	1	3	6	3.16	0.6
16	control versiones	x	14	6	7	10	7.3	0.4
17	levantamiento	x	15	1	2	4	2.16	0.25
18	instalación	15	16	4	5	8	5.3	0.4
19	instalación	16	17	2	3	5	3.16	0.25
20	diagrama de porte	x	18	1	2	4	2.16	0.25
21	carta+logo+info	18	19	1	2	3	2	0.1
22	Descripción	19	20	0	1	3	1.16	0.25
23	info tablero+co	18	21	2	3	6	3.3	0.4
24	misión y visión	x	22	1	2	4	2.16	0.25
25	Objetivos	22	23	0	1	3	1.16	0.25
26	Logo isotipo	22	24	1	2	5	2.3	0.4
27	Misión, visión	22 y 23	25	0	1	3	1.16	0.25
28	Forma jurídica	25	26	0	1	2	1	0.1
29	Localización	25	27	0	1	3	1.16	0.25
30	acción académica	x	28	1	2	4	2.16	0.25
31	bajar en equipo	28	29	2	3	5	3.16	0.25
32	crear idea física	x	30	1	2	4	2.16	0.25



11. Prototipado de la aplicación en coordinación con el profesor de FullStack

El prototipado de la aplicación no se llegó a coordinar entre los profesores y el profesor de programación descartó Mock-up y Wireframe y nos dijo que utilizaríamos Figma, siendo así lo hecho en Figma:

<https://www.figma.com/design/PUNSFszpfYbmDKa5xLgKOC/Sin-t%C3%ADtulo?node-id=3-275&t=11muavSOZtGFWfo1-1>

12. Carta de presentación del equipo

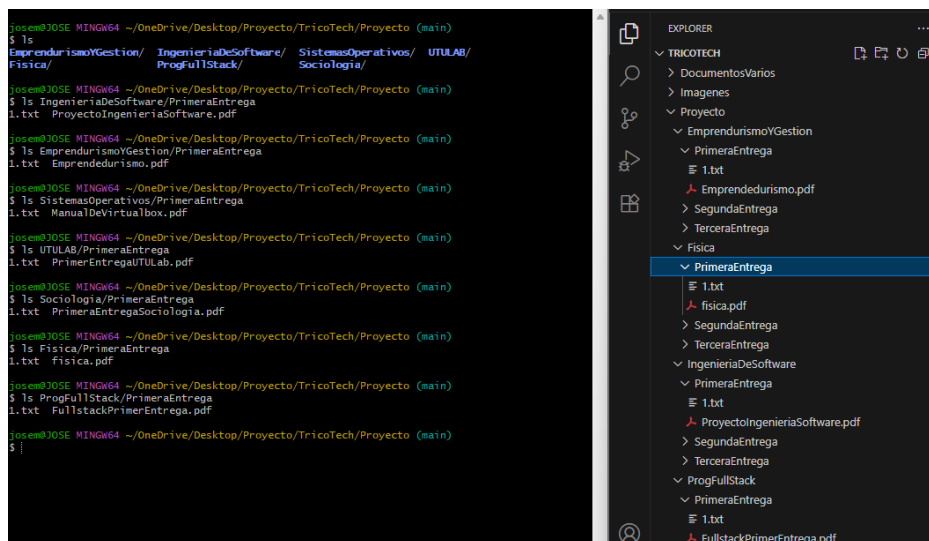
https://drive.google.com/file/d/1grOXmymNlic7UjPJ2cLtOCPIhd995xf4/view?usp=drive_link

13. Reglamento interno del equipo

https://drive.google.com/file/d/1O8MMVFr4tLZy-5nfqa6U4p2CQ42w_Fwf/view?usp=drive_link

14. Creación Repositorio privado en Git y subir

El repositorio ya fue creado y ya se encuentra TODO el proyecto subido al repositorio como se puede ver en la captura





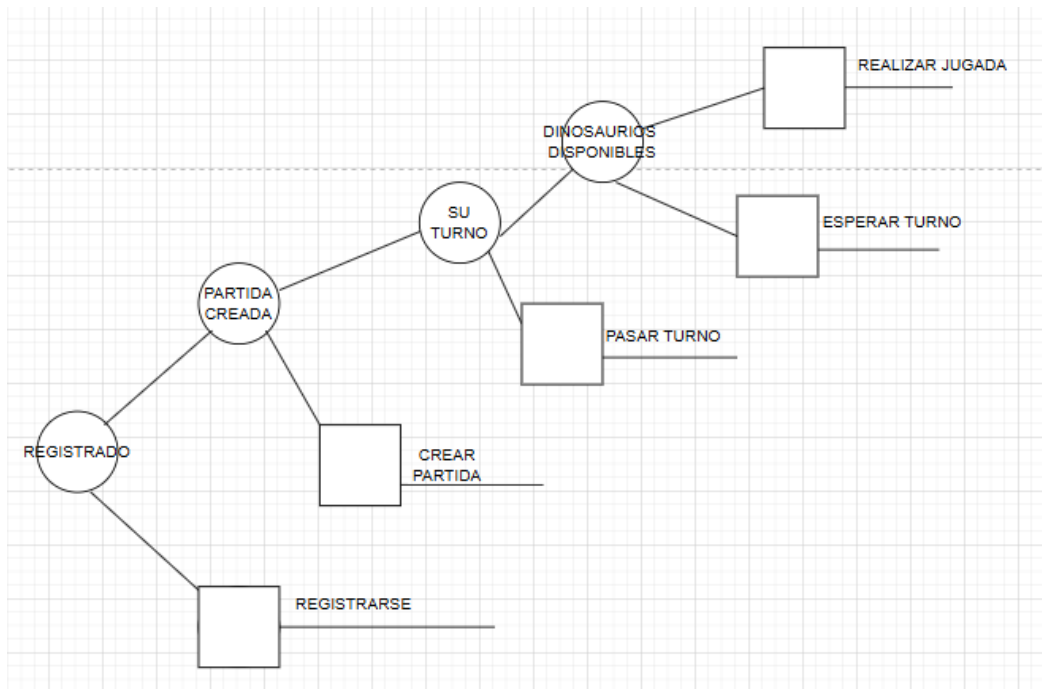
15. Tabla de decisiones

Las condiciones fueron: ¿Jugador registrado? ¿Partida creada? ¿Es turno del jugador? ¿Dinosaurios en draft disponibles?

Las acciones fueron:

- A. Registrarse
- B. Crear partida
- C. Realizar jugada
- D. Pasar turno

Árbol de decisiones suponiendo en un círculo como si fuera un SI (IF), mientras que el cuadrado significa ENTONCES (THEN).



$2^4=16$ reglas posibles, pero de esas 16 posibles reglas muchas son imposibles o redundantes como, por ejemplo:

De esas 16 combinaciones teóricas, no todas tienen sentido en el contexto del juego.

Ejemplos:

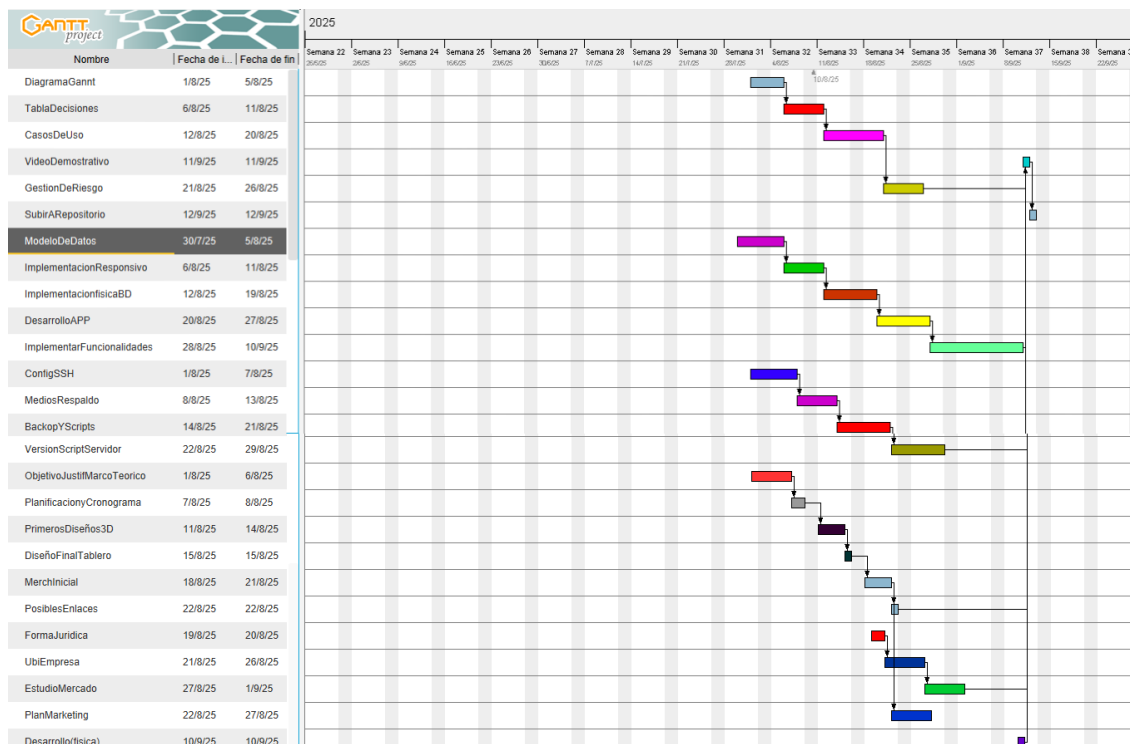
1. Si no hay jugador registrado, entonces no puede haber turno ni jugada, se simplifica a “hacer login”
2. Si no hay partida creada, da igual si hay turno o dinosaurios la única acción válida es “crear partida”
3. Si hay partida creada pero no es turno, la única acción lógica es “pasar turno”.

Quedaron esas 8 reglas útiles



Condiciones / Reglas	R1	R2	R3	R4	R5	R6	R7	R8
Jugador registrado	Sí	No	Sí	Sí	Sí	Sí	No	Sí
Partida creada	Sí	–	No	Sí	Sí	No	–	Sí
Es turno del jugador	Sí	–	–	No	Sí	–	–	No
Dinosaurios en draft disponibles	Sí	–	–	–	No	–	–	–
Acciones								
Registrarse	–	✓	–	–	–	–	✓	–
Crear partida	–	–	✓	–	–	✓	–	–
Realizar jugada	✓	–	–	–	–	–	–	–
Pasar turno	–	–	–	✓	–	–	–	✓

16. DiagramaGantt 2da entrega



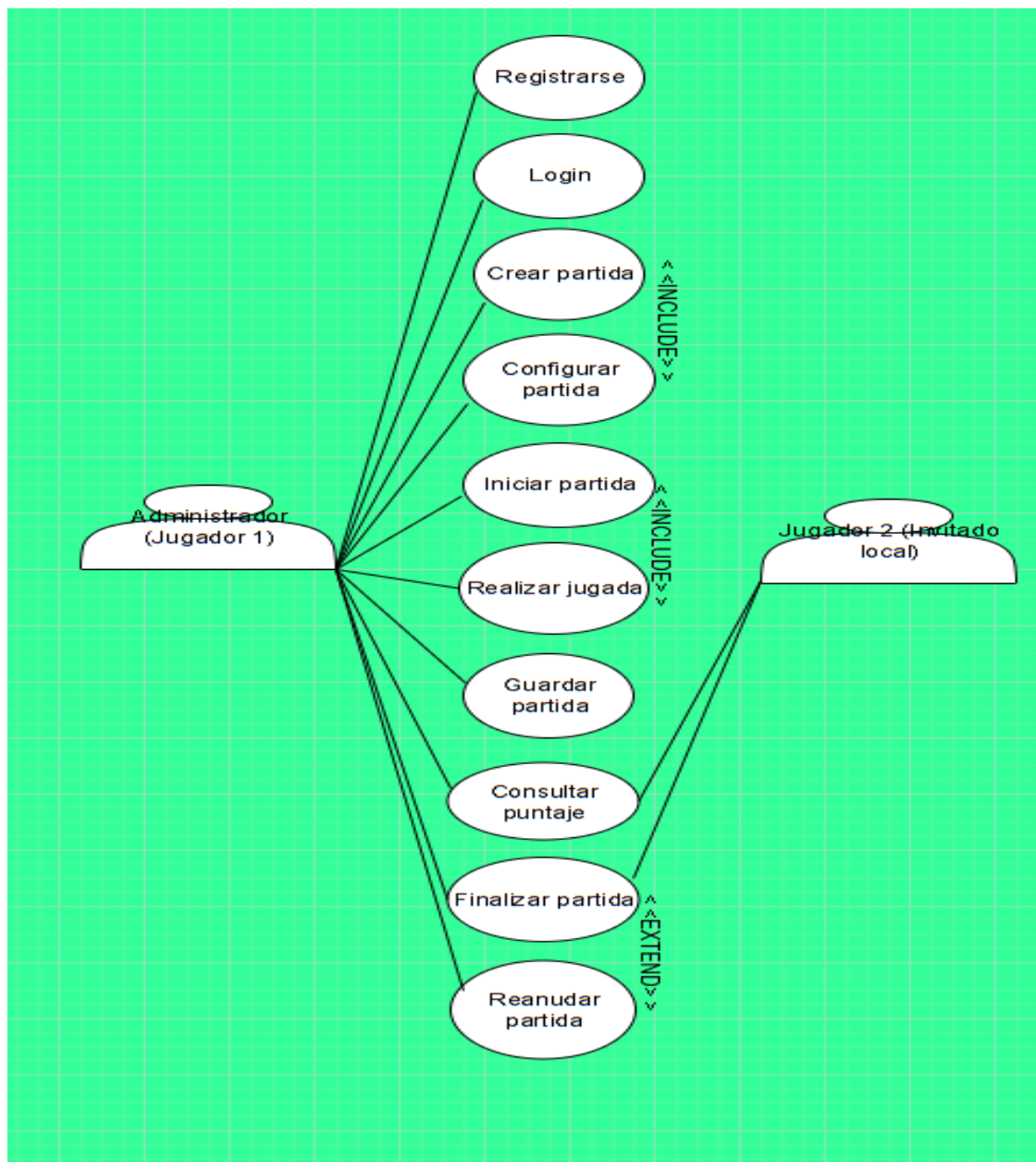
La entrega inicio el 01/08/2025 finaliza el dia 12/09/2025

(va adjunto el archivo del diagrama en GanntProyect)



17. Diagrama Casos de uso

Como por ahora la conexión con el juego va a ser mediante el uso de servidor local nos termina quedando el siguiente diagrama de casos de uso con 10 casos





18. Plantilla de casos de uso

CU01	Registrarse	
Descripción	El jugador crea una cuenta en el sistema.	
Precondición	El jugador no debe estar registrado previamente.	
Curso Normal	Acción Actor	Acción sistema
	1 - El jugador accede a la opción "Registrarse"	1 - El sistema solicita los datos (usuario, correo, contraseña, etc.).
	2 - El cliente ingresa los datos requeridos y presiona el botón "aceptar"	2 - El sistema valida los datos, crea la cuenta y notifica al jugador.
Curso alternativo	Acción Actor	Acción sistema
	1 - El cliente ingresa un N° de C.I. ya ingresado.	1 - El sistema despliega el mensaje "El usuario ya está ingresado en el sistema"
Postcondición	El jugador queda registrado en el sistema.	

CU02	Login	
Descripción	El jugador ingresa al sistema con sus credenciales.	
Precondición	El jugador debe estar registrado.	
Curso Normal	Acción Actor	Acción sistema
	1 -El jugador selecciona "Iniciar sesión."	
	2 - Ingresa usuario y contraseña.	2 - El sistema valida credenciales y permite el acceso
Curso alternativo	Acción Actor	Acción sistema
	1 - El cliente ingresa un N° de C.I. incorrecto	1 - El sistema despliega el mensaje "Error al encontrar credenciales"
Postcondición	El jugador queda autenticado en el sistema.	

CU03	Crear Partida	
Descripción	El administrador crea una nueva partida.	
Precondición	El jugador debe estar autenticado.	
Curso Normal	Acción Actor	Acción sistema
	1 -El administrador selecciona "Crear partida."	1-El sistema solicita los parámetros de la partida.
	2 - El administrador configura los parámetros.	2 - El sistema genera la partida.
Curso alternativo	Acción Actor	Acción sistema
	1 - Si falta información o hay algún error	1 - El sistema despliega el mensaje "Error al crear partida"
Postcondición	La partida queda registrada en estado Creada.	



CU04	Configurar partida	
Descripción	Se definen los parámetros de la partida.	
Precondición	La partida debe estar creada.	
Curso Normal	Acción Sistema	Acción actor
	1 -El sistema solicita parámetros de configuración (jugadores, reglas, etc.). 2 - El sistema valida y guarda la configuración.	1-El jugador ingresa la información.
Curso alternativo	Acción Actor	Acción sistema
	1 – El jugador ingresa parámetros inválidos	1 - El sistema despliega el mensaje “Parámetros inválidos”
Postcondición	La partida queda en estado Configurada	

CU05	Iniciar Partida	
Descripción	El administrador da comienzo a una partida configurada.	
Precondición	La partida debe estar configurada.	
Curso Normal	Acción Actor	Acción Sistema
	1 -El administrador selecciona “Iniciar partida.”	1-El sistema cambia el estado a “Iniciada.” 2- El sistema habilita jugadas
Curso alternativo	Acción Actor	Acción sistema
	1 – Si hubo un error en la configuración	1 - El sistema notificara al jugador
Postcondición	La partida pasa a estado <i>En curso</i> .	

CU06	Realizar jugada	
Descripción	Cada jugador realiza su turno siguiendo las reglas.	
Precondición	La partida debe estar en curso.	
Curso Normal	Acción Sistema	Acción Actor
	1 -El sistema indica el turno del jugador. 2-El sistema valida la jugada, actualiza el tablero y pasa al siguiente turno.	1-El jugador selecciona su acción.
Curso alternativo	Acción Actor	Acción sistema
	1 – El jugador hace una jugada invalida	1 - El sistema solicita otra acción
Postcondición	El tablero se actualiza con la jugada.	



CU07	Guardar partida	
Descripción	Se almacena el estado actual de la partida.	
Precondición	La partida debe estar en curso.	
Curso Normal	Acción Actor	Acción Sistema
	1 -El administrador selecciona " <i>Guardar partida.</i> "	1-El sistema almacena el estado actual en la base de datos.
Curso alternativo	Acción Actor	Acción sistema
	1 -El administrador selecciona " <i>Guardar partida.</i> "	1 - El sistema notifica que no hay espacio de almacenamiento y rechaza la accion
Postcondición	La partida queda en estado <i>Guardada</i> .	

CU08	Reanudar partida	
Descripción	Se retoma una partida previamente guardada.	
Precondición	Debe existir una partida guardada.	
Curso Normal	Acción Actor	Acción Sistema
	1 -El administrador selecciona " <i>Reanudar partida.</i> "	1-El sistema carga el estado almacenado.
		2- El sistema coloca la partida en estado <i>En curso</i> .
Curso alternativo	Acción Actor	Acción sistema
	1 -No existe partida guardada	1 - El sistema notifica que no hay ninguna partida guardada
Postcondición	La partida vuelve a estado <i>En curso</i> .	

CU09	Finalizar partida	
Descripción	Se da por concluida la partida.	
Precondición	La partida debe estar en curso	
Curso Normal	Acción Actor	Acción Sistema
	1 -El jugador selecciona " <i>Finalizar partida</i> "	1-El sistema cierra la partida.
		2- El sistema coloca la partida en estado <i>En curso</i> .
Curso alternativo	Acción Actor	Acción sistema
	1 -La partida ya estaba finalizada	1 - El sistema notifica que la partida ya estaba finalizada
Postcondición	La partida queda en estado <i>Finalizada</i> .	



CU10	Consultar puntaje	
Descripción	Los jugadores consultan los resultados de la partida.	
Precondición	La partida debe estar finalizada.	
Curso Normal	Acción Actor	Acción Sistema
	1 -Los jugadores visualizan los resultados.	1-El sistema muestra los puntajes obtenidos.
Curso alternativo	Acción Actor	Acción sistema
	1 -Los puntajes no se registraron correctamente	1 - El sistema despliega un mensaje de "Error al consultar los puntajes"
Postcondición	Los resultados quedan disponibles para análisis o registro.	

19. Diagrama de estados

Este diagrama de estados representa el ciclo de vida de una partida en el sistema.

Estados principales:

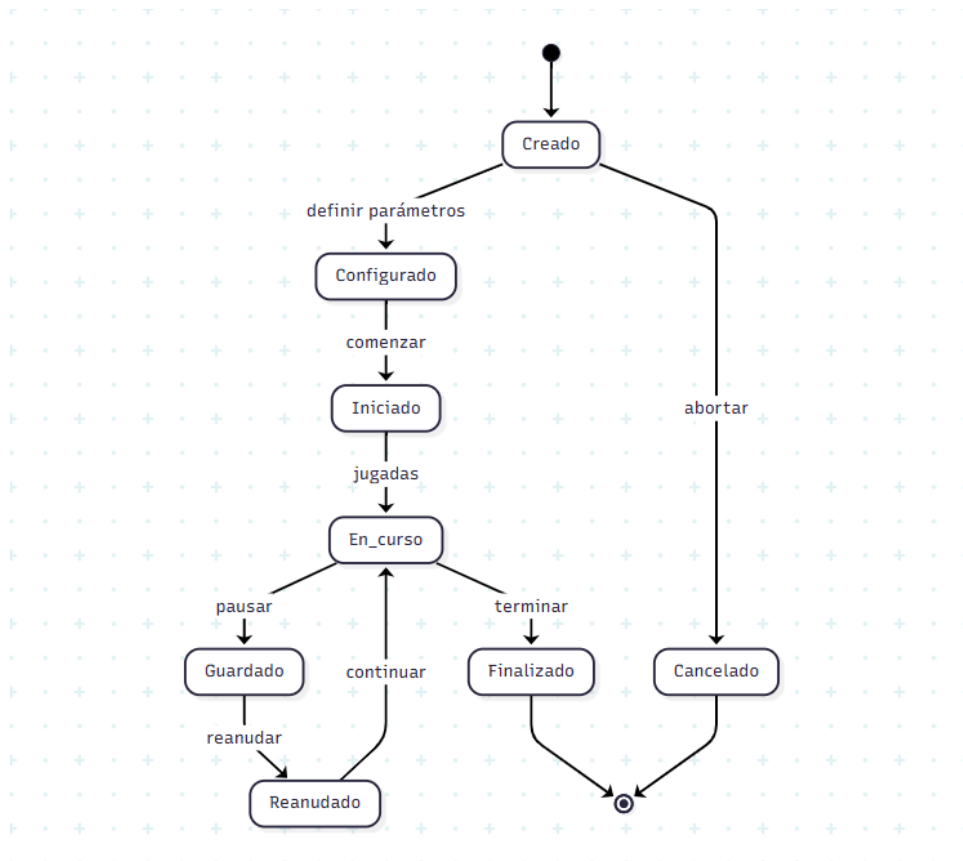
- **Creado:** Se genera una nueva partida por parte del administrador.
- **Configurado:** Se definen los parámetros de la partida (jugadores, reglas, etc.).
- **Iniciado:** Comienza oficialmente la partida.
- **En curso:** Estado activo donde los jugadores realizan jugadas.
- **Guardado:** La partida queda almacenada para reanudar más tarde.
- **Reanudado:** Una partida guardada vuelve a ponerse en curso.
- **Finalizado:** El juego termina y se muestran resultados.
- **Cancelado:** La partida se aborta antes de iniciarse o concluir.

Transiciones:

- Creado → Configurado: al definir parámetros.
- Creado → Cancelado: si se aborta antes de configurarla.
- Configurado → Iniciado: al comenzar el juego.
- Iniciado → En curso: cuando los jugadores realizan jugadas.
- En curso → Guardado: al pausar y almacenar la partida.



- En curso → Finalizado: cuando se decide terminar.
- Guardado → Reanudado: si se abre una partida previa.
- Reanudado → En curso: se continúa el juego.
- Finalizado / Cancelado → Fin: marca el cierre del proceso.



20. Estimaciones

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,23	1,08	1,00	1,04	1,10	



Tamaño DB (1), Complejidad (1,15), Restricciones tiempo ejecución (1), Capacidad análisis (1), Experiencia en la app (1), Experiencia en el lenguaje (1), Utilización herramientas software (0.91), Restricciones tiempo de desarrollo (1.04) → FAE = 1.08

Estimamos unas 6000 líneas de código sin los comentarios y espacios en blanco

$O=4000 \quad m=6000 \quad b=8000 \rightarrow (4000+4(6000)+8000) \% 6 = 6000$ Líneas esperadas

6 KLDC

Usamos el modo orgánico en el COCOMO intermedio

<i>Modo de desarrollo</i>	COCOMO Básico <i>a</i>	COCOMO Intermedio <i>A</i>	<i>b</i>	<i>c</i>	<i>d</i>
Orgánico	2.4	3.2	1.05	2.50	0.38
Semiacoplado	3.0		1.12		0.35
Empotrado	3.6	2.8	1.20		0.32

Esfuerzo(E) = $3.2 (6) 1.05 \times 1.08 = 22.67$

Tiempo(T) = $2.5 (22.67) 0.38 = 8.18$

Personal promedio(P) = $22.67 / 8.18 = 2.7$

Resultado final

- **Esfuerzo:** 23 persona-mes
- **Duración:** 9 meses
- **Personal promedio:** 3 personas