

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



Contenido

1. Relevamiento	4
1.1 Formulario relevamiento.....	4
1.2 Técnicas utilizadas	4
2. Estudio de factibilidades	4
2.1 Factibilidad Operativa	4
2.2 Factibilidad Técnica	5
2.3 Factibilidad Legal.....	6
3. Paradigma de conformación del equipo	6
4. Especificación de Requerimientos	7
Definiciones, acrónimos y abreviaturas	7
Perspectiva del producto.....	7
Funcionalidades del producto.....	7
Características de los usuarios.....	8
Alcance.....	8
Limitaciones	8
Suposiciones y dependencias.....	8
Requerimientos funcionales	9
Requerimientos no funcionales	9
Restricciones acerca del lenguaje de programación	10
5. Implementación del modelo de desarrollo	10
6. Lógica del sistema.....	11
7. Documentación de inicio y planificación del proyecto	12
8. Paradigma de conformación del equipo (Coordinación con UTULab) ...	12
9. Diagrama de Gantt primera entrega.....	13
10. Diagrama de PERT con GanttProject	14
11. Prototipado de la aplicación en coordinación con el profesor de FullStack	15
12. Carta de presentación del equipo	16
13. Reglamento interno del equipo	17
14. Creación Repositorio privado en Git y subir	18
15. Tabla de decisiones.....	18



16. DiagramaGantt 2da entrega.....	20
17. Diagrama Casos de uso	21
18. Plantilla de casos de uso	22
19. Diagrama de estados	25
20. Estimaciones	27
21. Riesgo y contingencia	28
22. Manual de instalación	36
23. Manual de usuario	37
24. Análisis Costo-Beneficio/ Factibilidad Económica.....	38
Tabla de finanzas y Plan Financiero (Emprendedurismo)	40
25.Tamaño punto por función.....	44
26. Analisis FODA ponderado.....	46
27.Diagramas Nassi-Shneiderman (N-S)	48
28. Diagrama de Gantt.....	52

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 como 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 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

Propósito

Este documento define los requerimientos funcionales, no funcionales, limitaciones y alcance para el desarrollo del proyecto final basado en el juego Draftosaurus. El sistema permitirá jugar modo de juego seguimiento, modo de juego digitalizado, registrar partidas y aprender sobre dinámica estratégica. Ideal para personas mayores de 8 años.

Definiciones, acrónimos y abreviaturas

- ERS: Especificación de Requerimientos de Software
- RF: Requerimiento Funcional.
- RNF: Requerimiento No Funcional
- GUI: Interfaz grafica
- PHP: Lenguaje de programación para backend
- JS: JavaScript
- App: Aplicación

Perspectiva del producto

El sistema es una aplicación web independiente accesible desde computadoras, celulares y tablets. No requiere instalación y será ejecutado localmente mediante un servidor PHP básico como XAMPP o algún similar directamente desde un entorno web. Mediante un diagrama de bloques quedaría algo así:

Cliente → Navegador web → GUI (HTML, CSS, JS) → Lógica del juego → Servidor local → Historial de partidas (.txt o BD)

Funcionalidades del producto

- Iniciar y reiniciar partidas
- Seleccionar cantidad de jugadores
- Tablero digital interactivo
- Sistema de puntuación automática
- Registro y consulta de historial de partidas

Características de los usuarios

El sistema está diseñado para ser utilizado principalmente por:

- **Jugadores:** Utilizarán la aplicación para el registro y cálculo de puntuación o para jugar en la versión digital. Usuarios que no necesitan tener conocimientos técnicos avanzados, que interactúan con el juego mediante una interfaz gráfica sencilla e intuitiva desde un navegador web

Administradores: Encargados de la configuración y mantenimiento del sistema, asegurando el correcto funcionamiento de la plataforma.

Todos los usuarios deben tener conocimientos básicos de informática, como navegar en internet, hacer clic, leer instrucciones en pantalla y utilizar una Computadora, Tablet o Celular

Alcance

El sistema consiste en una aplicación web para digitalizar y gestionar partidas del juego Draftosaurus, permitiendo registrar jugadores, iniciar partidas, seleccionar dinosaurios en el tablero digital, calcular puntajes y generar reportes para el seguimiento de los docentes. En cuanto al alcance del desarrollo, hay funcionalidades que están incluidas y otras que quedan fuera:

Dentro del alcance: Registro de jugadores, gestión de partidas (crear, iniciar y finalizar), tablero digital con selección de dinosaurios, cálculo de puntuaciones, guardado de partida (aunque el cliente no lo pidió, lo consideramos importante incluirlo), cumplimiento con los tiempos de entrega establecidos.

Fuera del alcance: Implementación de sonido o música en el juego (el cliente no lo solicitó), conexión con versiones físicas del juego, el modo de invierno.

Limitaciones

Como ya mencionaba en el alcance el tiempo es una limitación ya que no tenemos todo el tiempo del mundo para la finalización del proyecto. En las factibilidades mencionaba que no es una app, sino que se va jugar mediante web, que sea únicamente web es una limitación. Que sea solo compatible con navegadores actualizados y sistemas operativos mayores a Windows 7, se requiere entorno local para PHP como XAMPP 8.X. Capacidad de usuarios conectados es una limitación ya que es máximo 5 jugadores. El juego será únicamente el modo de verano, No incluye expansiones ni modos adicionales de Draftosaurus.

Suposiciones y dependencias

- Los usuarios tendrán acceso a un navegador actualizado
- Se usará XAMPP o entorno similar para ejecutar el sistema localmente
- Los usuarios tienen conocimiento mínimo de informática



Requerimientos funcionales

- RF1: El sistema permitirá crear una partida con entre 2 y 5 jugadores.
- RF2: El sistema mostrará un tablero digital con 6 islas (zonas) para colocar dinosaurios, según el modo verano de Draftosaurus.
- RF3: El sistema permitirá arrastrar y soltar cartas de dinosaurios únicamente en islas válidas, respetando las reglas del juego.
- RF4: El sistema calculará y mostrará automáticamente el puntaje final de cada jugador al terminar la partida.
- RF5: El sistema asignará los turnos de forma automática y secuencial entre los jugadores registrados.
- RF6: El sistema impedirá que un jugador coloque más de un dinosaurio del mismo tipo en una misma isla (excepto en la isla del Carnotauro).
- RF7: El sistema permitirá guardar el estado actual de la partida (jugadores, islas, cartas usadas y turno) para reanudarla después.
- RF8: El sistema mostrará un historial de partidas con fecha, duración, nombres de jugadores y puntajes obtenidos.
- RF9: El sistema permitirá a los usuarios registrarse e iniciar sesión con usuario y contraseña.
- RF10: El sistema validará que todos los jugadores hayan ingresado un nombre antes de comenzar la partida.
- RF11: El sistema permitirá reiniciar una partida en curso, siempre que el jugador confirme la acción.
- RF12: El sistema mostrará en tiempo real qué jugador está en turno y cuántas cartas le quedan.
- RF13: El sistema impedirá realizar movimientos fuera del turno del jugador activo.
- RF14: El sistema permitirá exportar el historial de partidas en formato CSV.
- RF15: El sistema restablecerá correctamente el estado del tablero al cargar una partida guardada.

Requerimientos no funcionales

- RNF1: La interfaz debe ser intuitiva y usable por personas mayores de 8 años sin instrucción previa.
- RNF2: El sistema debe cargarse en menos de 3 segundos en un dispositivo con conexión estable y navegador moderno.
- RNF3: La aplicación debe ser completamente responsive (adaptable a celulares, tablets y computadoras).
- RNF4: El sistema debe funcionar en los últimos dos años de versiones de Chrome, Firefox, Edge y Safari.
- RNF5: Las contraseñas de los usuarios deben almacenarse encriptadas usando un algoritmo seguro (ej. bcrypt).
- RNF6: El sistema no requiere conexión a internet y debe funcionar completamente en modo local (offline).
- RNF7: El diseño visual debe usar colores y tipografías legibles, con buen contraste para accesibilidad básica.
- RNF8: El sistema debe ejecutarse en equipos con al menos 4 GB de RAM y procesador básico (ej. Intel i3 o equivalente).
- RNF9: El sistema debe preservar los datos de partidas guardadas incluso si el



navegador se cierra inesperadamente.

RNF10: El sistema no debe mostrar errores técnicos al usuario; en su lugar, debe presentar mensajes amigables.

RNF11: El código fuente debe incluir comentarios mínimos en PHP y JavaScript para facilitar su revisión académica.

RNF12: El sistema no incluirá sonidos, animaciones complejas ni efectos visuales distractores.

RNF13: El sistema solo soportará el modo base de Draftosaurus (modo verano), sin expansiones ni reglas alternativas.

RNF14: El almacenamiento de datos (partidas, usuarios) se realizará localmente (archivos .txt o base de datos SQLite/MySQL en localhost).

RNF15: El sistema debe permitir su ejecución en entornos locales como XAMPP 8.x o similar, sin dependencias externas.

Formato de entrada y salida

Esto sirve para mostrar cómo se presentan los datos en pantalla o que espera el sistema:

Entrada: selección de número de jugadores (botones de 2 a 4)

Entrada: nombres de jugadores (input de texto)

Salida: tablero con fichas colocadas

Salida: puntajes finales al cerrar la partida

Salida: historial mostrado en tabla con nombre, fecha y puntos

Restricciones acerca del lenguaje de programación

El sistema se desarrollará únicamente con tecnologías web: HTML, CSS, JavaScript para el frontend, y PHP para funcionalidades de backend (como el historial de partidas). Se van a utilizar Frameworks para la facilitación de código y ahorro de tiempo de ser necesarios

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.

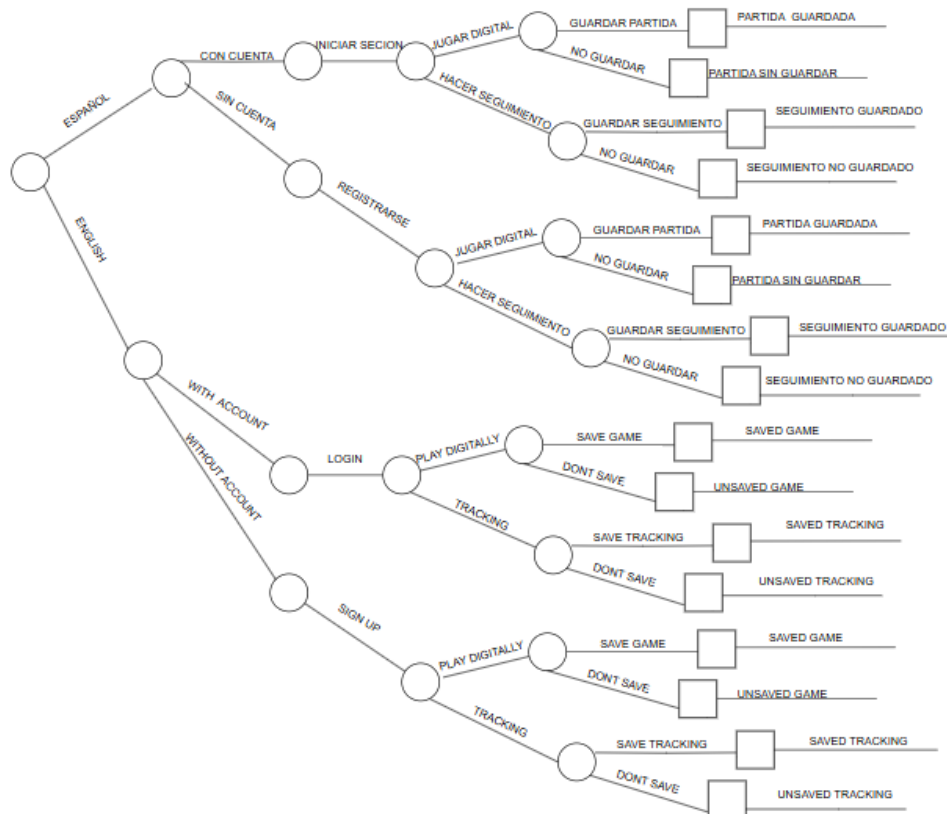
Ventajas de este modelo:

- 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

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)

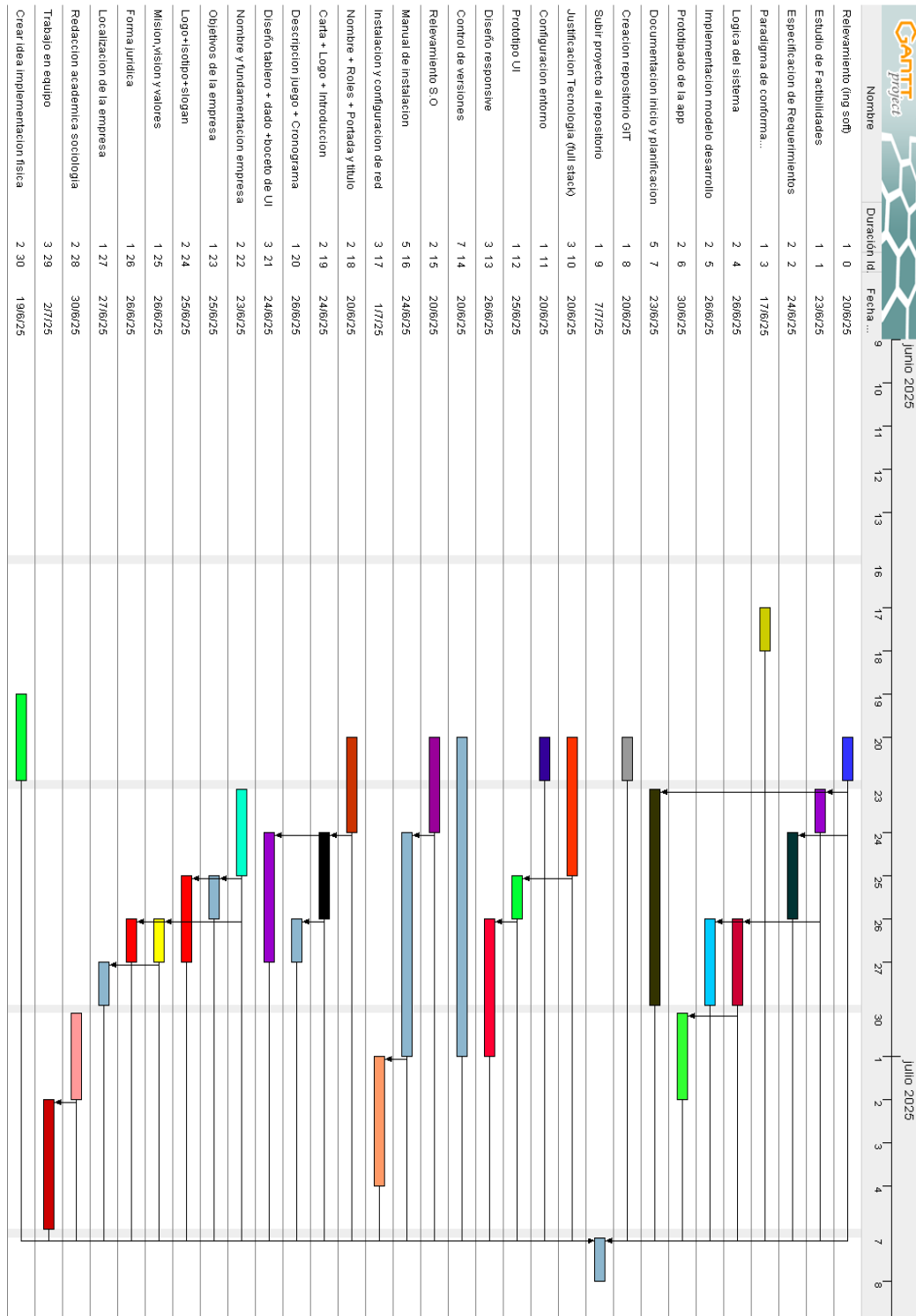
Como no hablamos aun de ningún paradigma en UTULab subo el mismo que para Ingeniería de Software. Este equipo de 3 personas está organizado por el Paradigma Descentralizado democrático (DD) porque no tenemos un líder permanente en si que tome solo las decisiones, se nombran coordinadores para tareas 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:



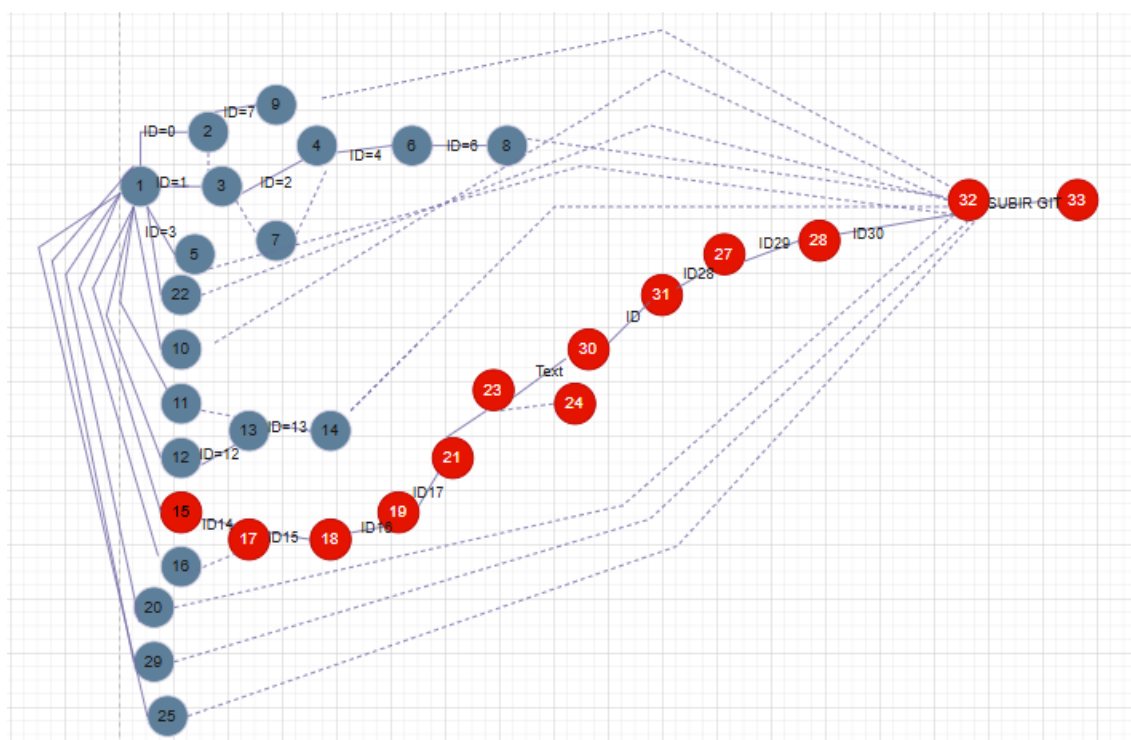
9. Diagrama de Gantt primera entrega

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

(va adjunto el archivo del diagrama en GanntProyect)



10. Diagrama de PERT con GanttProject



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

Actividad	Predecesor	ID	o	m	b	Te	σ^2
Relevamiento	x	0	0	1	3	1.16	0.25
Estudio fact	x	1	0	1	2	1	0.1
specificació	0 y 1	2	1	2	4	2.16	0.25
Paradigma	x	3	0	1	3	1.16	0.25
Logica	2	4	1	2	5	2.3	0.4
Modelo imp	1 y 2	5	1	2	6	2.5	0.6
Prototipado	4	6	1	2	5	2.3	0.4
ocumentaci	0	7	2	5	8	5	1
ear reposito	x	8	0	1	2	1	0.1
bir a reposito	TODAS	9	0	1	3	1.6	0.25
Justificaci	x	10	1	3	5	3	0.4
onfig entorn	x	11	0	1	3	1.16	0.25
Prototipo UI	10 y 11	12	4	6	9	6.16	0.6
eño respons	12	13	1	3	6	3.16	0.6
ontrol versio	x	14	6	7	10	7.3	0.4
levamiento	x	15	1	2	4	2.16	0.25
an instalaci	15	16	4	5	8	5.3	0.4
Instalacion	16	17	2	3	5	3.16	0.25
m+rol+porta	x	18	1	2	4	2.16	0.25
arta+logo+in	18	19	1	2	3	2	0.1
Descripci	19	20	0	1	3	1.16	0.25
ño tablero+c	18	21	2	3	6	3.3	0.4
e y fundame	x	22	1	2	4	2.16	0.25
Objetivos	22	23	0	1	3	1.16	0.25
Logo isotipo	22	24	1	2	5	2.3	0.4
Mision, visio	22 y 23	25	0	1	3	1.16	0.25
Forma juridic	23	26	0	1	2	1	0.1
Localizaci	25	27	0	1	3	1.16	0.25
ccion acade	x	28	1	2	4	2.16	0.25
abajo en equ	28	29	2	3	5	3.16	0.25
rear idea fisik	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 Muck-up y Wireframe y nos dijo que utilizáramos 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



TRICOTECH

Montevideo, 09 de Sept de 2025

Instituto Tecnológico de Informática

Presente.

A continuación, los alumnos de tercero 3MG del turno Nocturno del Instituto Tecnológico de Informática nos presentamos ante usted, con el fin de informar la creación del grupo TRICOTECH. Los correspondientes integrantes con sus roles son los siguientes:

A continuación, se detalla dicha integración y roles del grupo:

ROL	C.I	APELLIDO	NOMBRE	E-MAIL
Coordinador	5.484.815-1	Vanrell	Fabrizio	fabriciovanrell2@gmail.com
Subcoordinador	5.214.522-2	Madruga	José	josemadruga241999@gmail.com
Integrante 1	5.297.422-3	Benítez	Facundo	facuhtb98@gmail.com

Por contacto al correo: tricoTech@protonmail.com

Firmas:


COORDINADOR


SUBCOORDINADOR


INTEGRANTE 1



~~Vanrell~~, Fabrizio

COORDINADOR



Madruga, José

SUBCOORDINADOR



Benítez, Facundo

INTEGRANTE 1

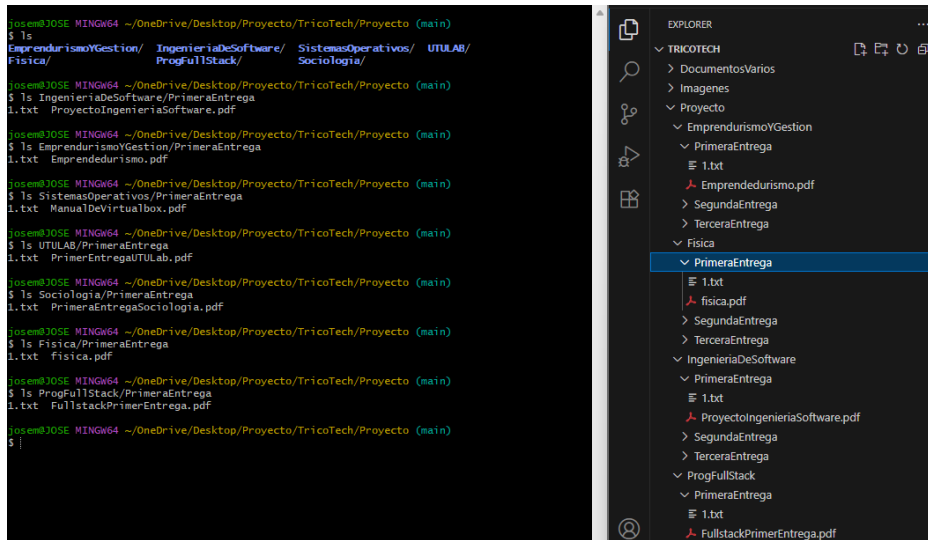


13. Reglamento interno del equipo

1. **Respeto Mutuo:** Todos los miembros del equipo se tratarán con respeto, cortesía y profesionalismo en todo momento, evitando cualquier forma de agresión verbal, personal o indirecta.
2. **Compromiso:** Cada miembro se compromete al 100% con las tareas asignadas y el éxito del proyecto.
3. **Incumplimiento Grave de Tareas:** Un miembro podrá ser excluido si, tras una advertencia formal del líder y un plazo de subsanación, persiste en el incumplimiento grave y reiterado de sus tareas asignadas, afectando el progreso del proyecto.
4. **Falta de Compromiso Manifiesta:** La falta de compromiso evidente, demostrada por la inactividad prolongada sin justificación como lo son 3 días sin comunicación o avance, la no participación en reuniones o la evasión sistemática de responsabilidades, podrá ser motivo de exclusión.
5. **Ausencia Injustificada Prolongada:** La ausencia a [Ej. 3] reuniones formales consecutivas sin justificación previa y aceptada por el equipo, o una inactividad total de [Ej. 7 días] sin aviso o mensaje excusándose, podrá ser considerada abandono del proyecto y llevar a la exclusión.
6. **Notificación Formal:** La exclusión de un miembro será comunicada de manera formal y por escrito al miembro afectado, con copia a los docentes/tutores del proyecto, si aplica.
7. **Cumplimiento de plazos y calidad en las entregas:** Los proyectos deben entregarse en los tiempos acordados y con los estándares de calidad definidos con el cliente
8. **Evitar conflictos personales dentro del equipo:** Si hay una diferencia con un compañero, resolverla hablando con madurez y sin afectar el trabajo del grupo.
9. **Uso responsable del repositorio compartido (GitHub):** No subir archivos innecesarios ni modificar código de otros sin coordinación previa.
10. **Atender fallos con actitud constructiva:** Si algo no funciona, se prueba en conjunto y se aprende del error, sin culpas ni excusas
11. **Expulsión por votos:** Si la mayor parte del grupo vota por la expulsión de algún miembro ese miembro será expulsado en el acto con efecto inmediato

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 ser en la captura



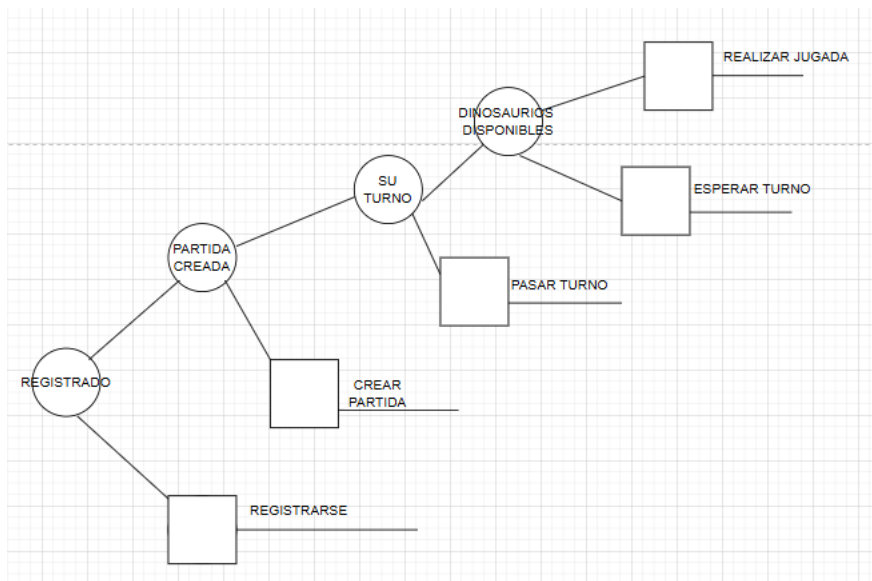
15. Tabla de decisiones

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).





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

Quedaron esas 8 reglas útiles

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

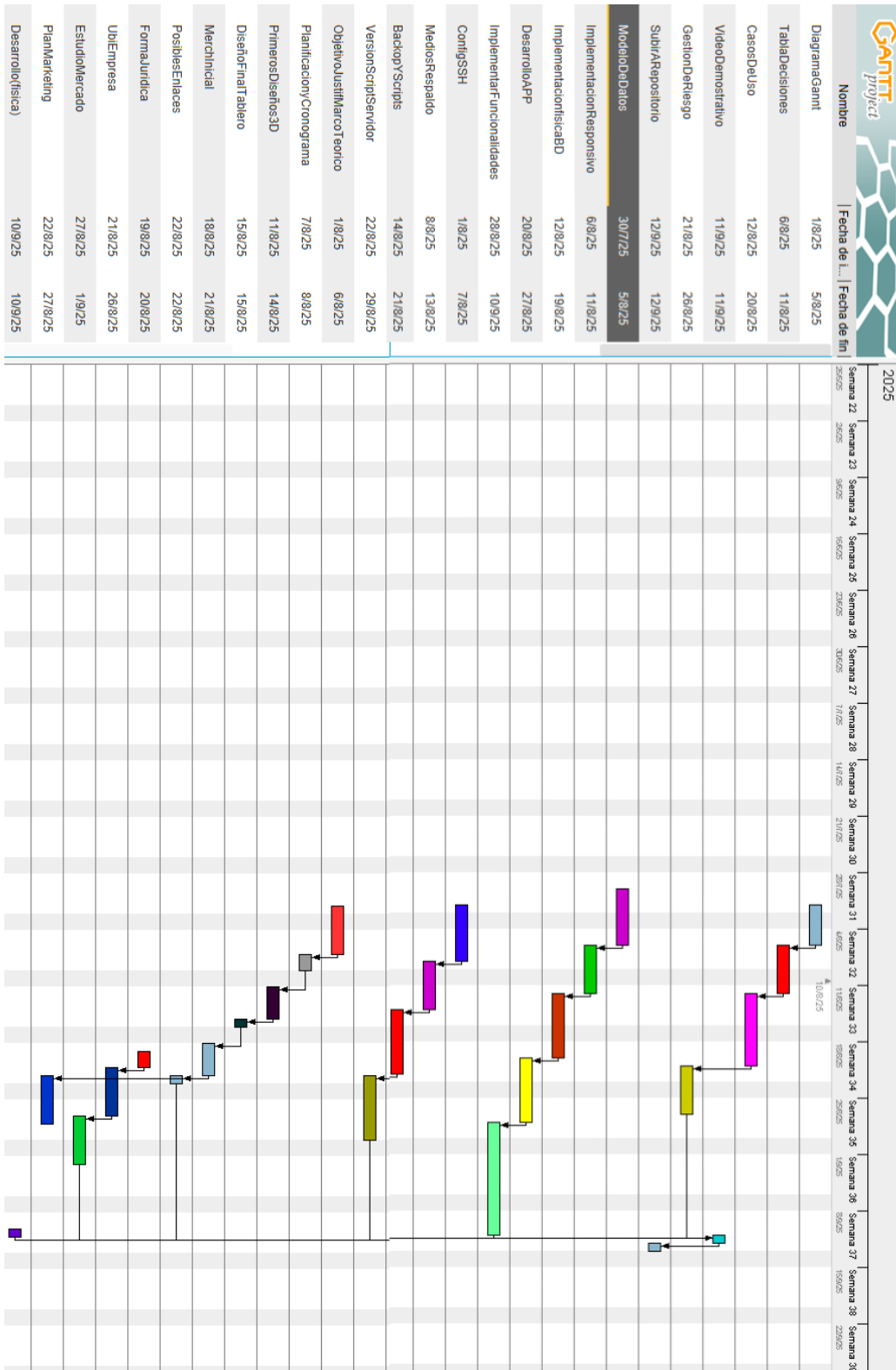
De esas 16 combinaciones teóricas, no todas tienen sentido en el contexto del juego.

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".



16. DiagramaGantt 2da entrega

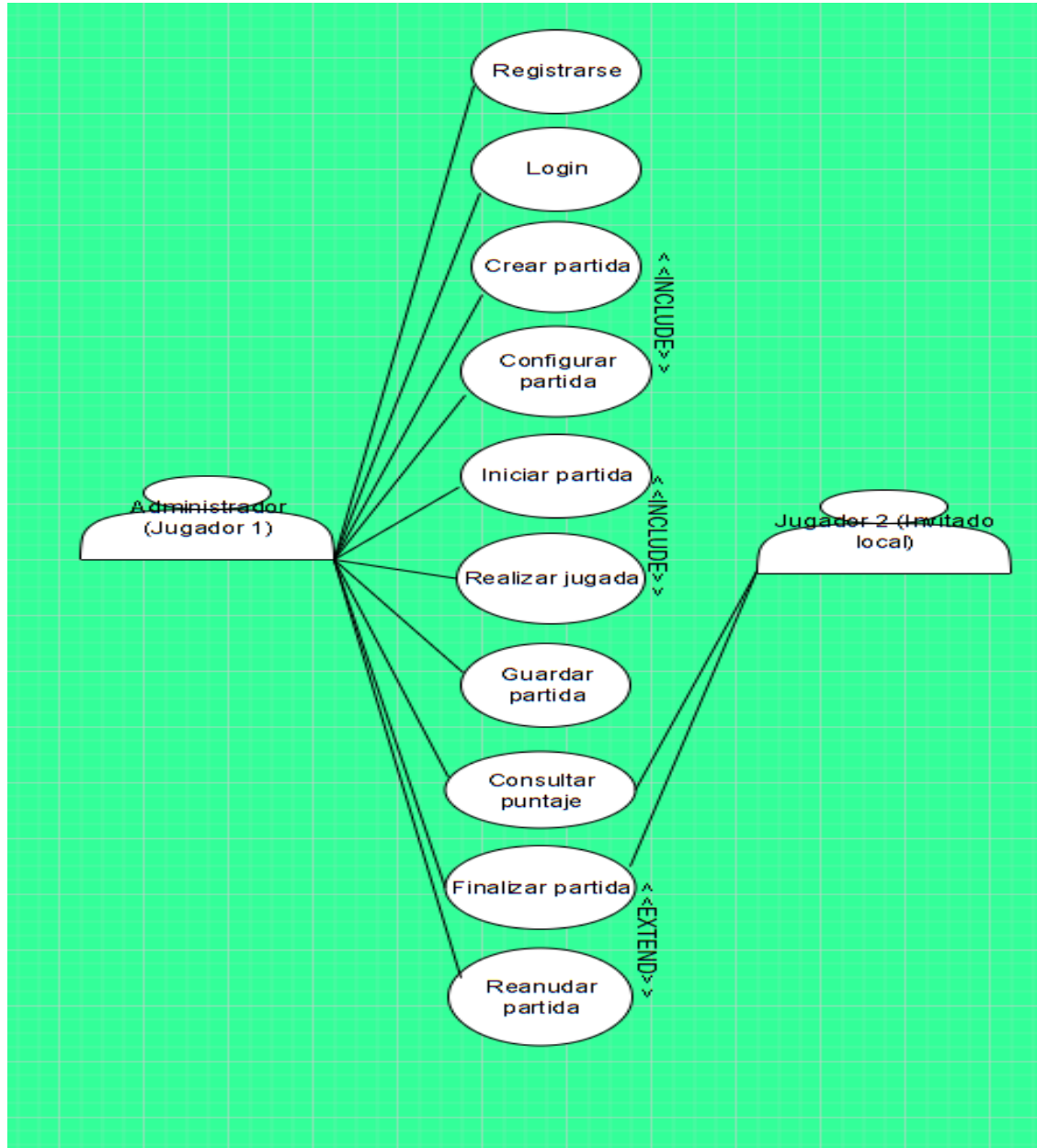


La entrega inicio el 01/08/2025 finaliza el dia 12/09/2025

(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.).	1-El jugador ingresa la información.
	2 - El sistema valida y guarda la configuración.	
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.	1-El jugador selecciona su acción.
	2-El sistema valida la jugada, actualiza el tablero y pasa al siguiente turno.	
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 "Guardar partida."	1-El sistema almacena el estado actual en la base de datos.
Curso alternativo	Acción Actor	Acción sistema
	1 -El administrador selecciona "Guardar partida."	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 "Reanudar partida."	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 "Finalizar partida"	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

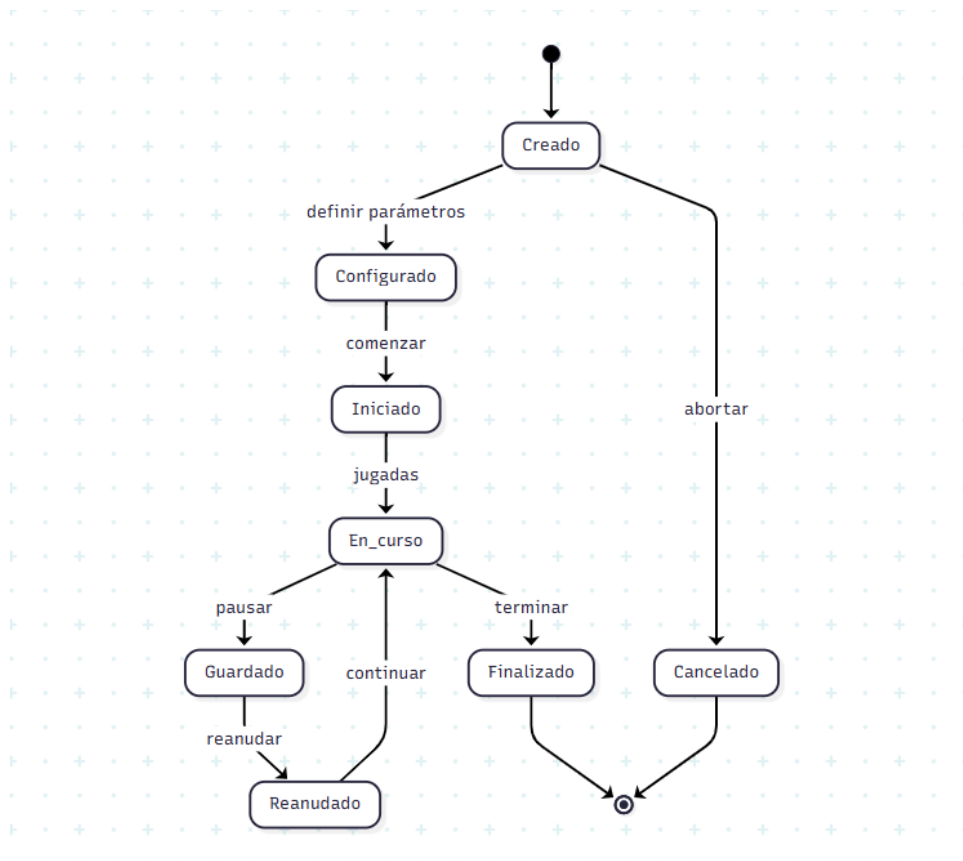
Este diagrama de estados representa el ciclo de vida de una partida en el sistema.

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	1,00	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

Modo de desarrollo	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



21. Riesgo y contingencia

IMPACTO

	1	2	3	4	5	PRO
1	1	2	3	4	5	BA
2	2	4	6	8	10	BI
3	3	6	9	12	15	LI
4	4	8	12	16	20	DAD
5	5	10	15	20	25	

BAJO

MEDIO

ALTO

EXTREMO

Hoja de información de riesgo

Riesgo ID: 01 **Fecha:** 15/10/2025 **Probabilidad:** 2 **Impacto:** 4

Descripción:

Ausencia de un integrante por razones mayores, ya sea por enfermedad o algún problema familiar

Refinamiento / Contexto:

El equipo está formado por solo 3 personas. Si uno no puede participar durante la recta final (antes del 3/11 o la defensa del 11/11), el resto debe asumir su carga, lo que nos puede generar retrasos y/o errores.

Mitigación / Monitoreo:

- Documentación constante en GitHub (commits diarios, README actualizado).
- Reuniones semanales obligatorias por Discord.
- Redistribuir tareas si es necesario.

Manejo / Plan de contingencia / Disparador:

- **Plan de contingencia:** El resto del equipo asume temporalmente las tareas del ausente usando la documentación como guía.
- **Disparador:** Ausencia confirmada por más de 2 días hábiles.

Estado:

En mitigación

Originador: José Madruga **Asignado:** Todo el equipo



Hoja de información de riesgo			
Riesgo ID: 02 Fecha: 17/10/2025 Probabilidad: 2 Impacto: 5			
Descripción:			
Cambios tardíos en requerimientos del proyecto. Puede ser agregar cosas o rehacer otras			
Refinamiento / Contexto:			
UTULab podría solicitar ajustes en reglas, modos de juego o funcionalidades a último momento, obligando a rehacer código o diseño ya validado.			
Mitigación / Monitoreo:			
<ul style="list-style-type: none"> - Validación de todos los requerimientos con UTULab antes de la 2.^a entrega. - Uso del prototipo en Figma como referencia visual aprobada. - Registro de cambios en documento compartido. 			
Manejo / Plan de contingencia / Disparador:			
<ul style="list-style-type: none"> - Plan de contingencia: Aplicar gestión formal de cambios; si es crítico, entregar funcional y posponer ajustes no esenciales. - Disparador: Notificación oficial de UTULab sobre modificación de requisitos. 			
Estado:			
En mitigación			
Originador: Fabricio Vanrell		Asignado: Fabricio Vanrell	

Hoja de información de riesgo			
Riesgo ID: 03 Fecha: 01/11/2025 Probabilidad: 2 Impacto: 3			
Descripción:			
Fallos en XAMPP o entorno local de la UTU			
Refinamiento / Contexto:			
El juego se desarrolla en XAMPP 8.x. Si falla (no inicia, errores de PHP), no se puede probar ni entregar una versión funcional.			
Mitigación / Monitoreo:			
<ul style="list-style-type: none"> - Todos usan la misma versión de XAMPP. - Manual de instalación en el repositorio. - Pruebas en servidor escolar como respaldo. 			
Manejo / Plan de contingencia / Disparador:			
<ul style="list-style-type: none"> - Plan de contingencia: Usar servidor de la UTU o reinstalar XAMPP desde cero con guía documentada. - Disparador: Imposibilidad de ejecutar el juego local por más de 1 día. 			
Estado:			
En mitigación			
Originador: Facundo Benítez		Asignado: Facundo Benítez	



Hoja de información de riesgo			
Riesgo ID: 04	Fecha: 01/11/2025	Probabilidad: 5	Impacto: 3
Descripción:			
Retraso por sobrecarga académica + trabajo + procrastinacion			
Refinamiento / Contexto:			
Tenemos múltiples entregas en noviembre. Esto reduce el tiempo disponible para dedicarle solo al proyecto			
Mitigación / Monitoreo:			
- Diagrama de Gantt.			
- Revisión semanal del avance vs plan.			
- Priorización de funcionalidades esenciales			
Manejo / Plan de contingencia / Disparador:			
- Plan de contingencia: Entregar versión mínima funcional y completar en iteración posterior.			
- Disparador: Atraso acumulado de más de 3 días en tareas críticas.			
Estado:			
En mitigación			
Originador:	José Madruga	Asignado:	José Madruga

Hoja de información de riesgo			
Riesgo ID: 05	Fecha: 10/10/2025	Probabilidad: 3	Impacto: 4
Descripción:			
Diseño no responsive en celulares o tablets.			
Refinamiento / Contexto:			
El juego debe ser jugable en cualquier dispositivo. Si no es responsive, se pierde usabilidad y afecta la evaluación.			
Mitigación / Monitoreo:			
-			
- Pruebas en múltiples dispositivos en cada iteración.			
- Feedback de compañeros con distintos celulares.			
Manejo / Plan de contingencia / Disparador:			
- Plan de contingencia: Simplificar interfaz o usar librería UI probada.			
- Disparador: Reporte de mal funcionamiento en dispositivos móviles.			
Estado:			
En mitigación			
Originador:	Fabrizio Vanrell	Asignado:	Fabrizio Vanrell



Hoja de información de riesgo			
Riesgo ID: 06	Fecha: 03/10/2025	Probabilidad: 3	Impacto: 3
Descripción:			
Falta de experiencia en PHP/JavaScript para resolver un problema.			
Refinamiento / Contexto:			
El equipo está aprendiendo. Algunos conceptos (sesiones, CORS validación) pueden bloquear el desarrollo.			
Mitigación / Monitoreo:			
<ul style="list-style-type: none"> - Estudio conjunto de temas difíciles. - Consulta al profe de programación - Uso de código de ejemplo validado. 			
Manejo / Plan de contingencia / Disparador:			
<ul style="list-style-type: none"> - Plan de contingencia: Simplificar funcionalidad o pedir ayuda externa puntual. - Disparador: Bloqueo del equipo sin solución momentánea. 			
Estado:			
En mitigación			
Originador: Facundo Benítez		Asignado: Facundo Benítez	

Hoja de información de riesgo			
Riesgo ID: 07	Fecha: 10/10/2025	Probabilidad: 1	Impacto: 4
Descripción:			
No lograr entregar funcionalidad mínima a tiempo.			
Refinamiento / Contexto:			
Si no se entrega al menos el MVP (modo seguimiento funcional, draft, puntuación), la nota se verá afectada.			
Mitigación / Monitoreo:			
<ul style="list-style-type: none"> - Dividir tareas en partes pequeñas y sencillas. - Validar funcionalidades críticas primero. - Lista de chequeo de MVP. 			
Manejo / Plan de contingencia / Disparador:			
<ul style="list-style-type: none"> - Plan de contingencia: Entregar versión mínima funcional - Disparador: Falta de progreso en funcionalidad crítica 			
Estado:			
En mitigación			
Originador: José Madruga		Asignado: Todo el equipo	



Hoja de información de riesgo			
Riesgo ID:	08	Fecha:	01/11/2025
		Probabilidad:	1
		Impacto:	4
Descripción:			
Falta de preparación del equipo para responder preguntas técnicas en la defensa.			
Refinamiento / Contexto:			
En la defensa del 11/11, vamos a defender el proyecto si no estamos preparados nos va a afectar mucho la nota final			
Mitigación / Monitoreo:			
- Reunión de ensayo de defensa programada para el 8/11.			
- Lista de preguntas probables con respuestas consensuadas.			
- Repaso conjunto de toda la documentación.			
Manejo / Plan de contingencia / Disparador:			
- Plan de contingencia: Todos hablar un poco sobre cada tema			
- Disparador: Día de la defensa (11/11).			
Estado:			
Identificado			
Originador: José Madruga			
Asignado: Todo el equipo			

Hoja de información de riesgo			
Riesgo ID:	09	Fecha:	01/11/2025
		Probabilidad:	2
		Impacto:	3
Descripción:			
No haber probado el juego en todos los navegadores soportados (Chrome, Firefox, Safari, Edge).			
Refinamiento / Contexto:			
El juego puede funcionar en Chrome pero fallar en Safari por diferencias en motores JS/CSS. Esto afecta la factibilidad técnica declarada.			
Mitigación / Monitoreo:			
- Programar prueba cruzada antes del 3/11.			
- Usar herramientas de desarrollo del navegador para simular.			
Manejo / Plan de contingencia / Disparador:			
- Plan de contingencia: Priorizar compatibilidad con Chrome y Firefox; aclarar en documentación que otros navegadores pueden tener limitaciones.			
- Disparador: Día de la entrega (3/11).			
Estado:			
Identificado			
Originador: Facundo Benítez			
Asignado: Facundo Benítez			



Hoja de información de riesgo			
Riesgo ID:	10	Fecha:	01/11/2025
Probabilidad:	3	Impacto:	3
Descripción:			
Pérdida de motivación del equipo en la recta final por agotamiento.			
Refinamiento / Contexto:			
Después de meses de trabajo, es común bajar el ritmo justo antes de entregar, lo que puede generar errores o descuidos.			
Mitigación / Monitoreo:			
<ul style="list-style-type: none"> - Reuniones breves pero frecuentes. - Celebraremos avances (ej.: "hoy terminamos el draft"). - Dividir tareas equitativamente. 			
Manejo / Plan de contingencia / Disparador:			
<ul style="list-style-type: none"> - Plan de contingencia: Tomar un día de descanso corto si el agotamiento es alto; reorganizar tareas para que todos tengan algo motivador. - Disparador: Notar que nadie avanza durante 2 días seguidos. 			
Estado:			
En mitigación			
Originador:	Fabrizio Vanrell	Asignado:	Todo el equipo

Hoja de información de riesgo			
Riesgo ID:	11	Fecha:	25/10/2025
Probabilidad:	3	Impacto:	4
Descripción:			
No haber actualizado la documentación después de los cambios de la 2. ^a a la 3. ^a entrega.			
Refinamiento / Contexto:			
Si la documentación no refleja lo implementado (ej.: nuevos casos de uso, cambios en el árbol de decisiones), genera inconsistencia en la defensa.			
Mitigación / Monitoreo:			
<ul style="list-style-type: none"> - Revisión cruzada de toda la doc antes del 3/11. - Actualización simultánea de código y documentación. 			
Manejo / Plan de contingencia / Disparador:			
<ul style="list-style-type: none"> - Plan de contingencia: Priorizar actualización de secciones clave: casos de uso, diagrama de estados, Gantt, estimaciones. - Disparador: Entrega final (3/11). 			
Estado:			
Identificado			
Originador:	José Madruga	Asignado:	Todo el equipo



Hoja de información de riesgo			
Riesgo ID: 12	Fecha: 01/11/2025	Probabilidad: 2	Impacto: 2
Descripción:			
Errores tipográficos o de formato en la documentación final.			
Refinamiento / Contexto:			
Errores menores dan mala impresión y sugieren descuido, aunque el contenido sea sólido.			
Mitigación / Monitoreo:			
- Revisión cruzada entre integrantes.			
- Uso de corrector ortográfico.			
- Exportar PDF con vista previa.			
Manejo / Plan de contingencia / Disparador:			
- Plan de contingencia: Corregir errores menores hasta el último momento antes de entregar.			
- Disparador: Entrega final (3/11).			
Estado:			
En mitigación			
Originador:	Fabrizio Vanrell	Asignado:	Fabrizio Vanrell

Hoja de información de riesgo			
Riesgo ID: 13	Fecha: 01/11/2025	Probabilidad: 2	Impacto: 3
Descripción:			
No tener un plan para dividir el tiempo de exposición entre los 3 integrantes.			
Refinamiento / Contexto:			
En la defensa, si uno habla todo el tiempo, parece que los otros no participaron. Si no hay plan, se interrumpen o se repiten.			
Mitigación / Monitoreo:			
- Asignar temas específicos a cada integrante (José → gestión, Fabrizio → diseño, Facundo → código).			
- Ensayar con cronómetro.			
Manejo / Plan de contingencia / Disparador:			
- Plan de contingencia: Usar un guion escrito con turnos claros.			
- Disparador: Día de la defensa (11/11).			
Estado:			
Identificado			
Originador:	José Madruga	Asignado:	Todo el equipo



Hoja de información de riesgo			
Riesgo ID: 14	Fecha: 20/10/2025	Probabilidad: 3	Impacto: 5
Descripción:			
Bugs críticos no detectados antes de la entrega (ej. puntuación incorrecta)			
Refinamiento / Contexto:			
Un bug en la lógica del draft o puntuación hace que el juego no funcione correctamente, afectando la evaluación.			
Mitigación / Monitoreo:			
<ul style="list-style-type: none"> - Pruebas manuales por cada integrante. - Lista de chequeo de funcionalidades críticas. - Testeo cruzado (cada uno prueba lo que hizo otro). 			
Manejo / Plan de contingencia / Disparador:			
<ul style="list-style-type: none"> - Plan de contingencia: Corregir bugs críticos antes de entregar; dejar bugs menores para después. - Disparador: Bug que impide jugar o finalizar partida. 			
Estado:			
En mitigación			
Originador:	Facundo Benítez	Asignado:	Facundo Benítez

Hoja de información de riesgo			
Riesgo ID: 15	Fecha: 23/10/2025	Probabilidad: 2	Impacto: 3
Descripción:			
Falta de comunicación efectiva entre integrantes.			
Refinamiento / Contexto:			
Trabajo descentralizado. Sin comunicación clara, hay duplicación de esfuerzos o tareas olvidadas.			
Mitigación / Monitoreo:			
<ul style="list-style-type: none"> - Reuniones fijas semanales. - Uso de WhatsApp para avisos urgentes. - Registro de decisiones en actas compartidas. 			
Manejo / Plan de contingencia / Disparador:			
<ul style="list-style-type: none"> - Plan de contingencia: Reunión de emergencia para realinear. Asignar coordinador temporal. - Disparador: Confusión recurrente sobre responsabilidades. 			
Estado:			
En mitigación			
Originador:	José Madruga	Asignado:	Todo el equipo

Se identificaron 15 riesgos críticos relacionados con el desarrollo, entrega y defensa del proyecto Draftosaurus. De ellos, 11 están en mitigación activa (con acciones ya implementadas), 4 están identificados (requieren acción inmediata antes del 3/11)

El equipo revisará esta lista en cada reunión semanal hasta la defensa del 11/11.
Cualquier cambio en probabilidad, impacto o estado será actualizado en el repositorio
Hasta el 3/11: enfocarse en los riesgos identificados (R-08, R-09, R-11, R-13): preparar defensa, probar en navegadores, actualizar documentación y ensayar exposición.
Del 4/11 al 11/11: simular la defensa completa, corregir últimos bugs y asegurar copia de respaldo en USB y GitHub.
Post-defensa: todos los riesgos se marcarán como “Cerrado”, y se archivará la gestión como lección aprendida y vamos a aprobar el año.

22. Manual de instalación

En este documento vamos a describir cómo instalar y configurar la API REST de Draftosaurus en un entorno local con XAMPP.

Requisitos previos:

- Windows 10/11
- XAMPP 8.0 o superior (incluye Apache, MySQL y PHP 8.0+)
- MySQL Workbench (opcional)
- Postman (opcional, para pruebas)

Paso 1: Descargar el proyecto

Descargue la carpeta del proyecto llamada “juegoDraftosaurus”.
Colóquela en la carpeta htdocs de XAMPP: C:\xampp\htdocs\juegoDraftosaurus

Paso 2: Configurar la base de datos

- Abrir XAMPP Control Panel e inicie Apache y MySQL.
- Abrir su navegador y vaya a: <http://localhost/phpmyadmin>
- Cree una nueva base de datos llamada: draftosaurus
- Seleccione la base de datos y vaya a la pestaña “SQL”.
- Pegue y ejecute el script SQL proporcionado (draftosaurus.sql).

Paso 3: Configurar Apache

Abra el archivo: C:\xampp\apache\conf\httpd.conf
Busque la línea: #LoadModule rewrite_module modules/mod_rewrite.so
y elimine el símbolo # al inicio.
Al final del archivo, agregue: <Directory "C:/xampp/htdocs/juegoDraftosaurus">
AllowOverride All Require all granted</Directory>
Guarde el archivo y reinicie Apache en XAMPP.

Paso 4: Verificar la instalación

Abra su navegador y visite: <http://localhost/juegoDraftosaurus>
Debería ver la página de documentación.
Pruebe el endpoint: <http://localhost/juegoDraftosaurus/partidas>
Debe mostrar: []
La instalación se ha completado correctamente.

23. Manual de usuario

En este documento vamos a describir cómo utilizar la API REST de Draftosaurus para gestionar partidas del juego.

Acceso a la API:

La API está disponible en:

<http://localhost/juegoDraftosaurus>

Endpoints principales

Partidas

-GET /partidas → Lista todas las partidas

-GET /partidas/{id} → Obtiene una partida específica

-POST /partidas → Crea una nueva partida

Ejemplo de datos para crear una partida: { "nombre_jugador1": "Ana",
"nombre_jugador2": "Luis" }

Bolsa inicial

GET /bolsa/{id_partida} → Obtiene la composición de la bolsa

POST /bolsa → Crea la bolsa inicial (48 dinosaurios)

Ejemplo: { "id_partida": 1, "trex_cantidad": 8, "especie_a_cantidad": 8,
"especie_b_cantidad": 8, "especie_c_cantidad": 8, "especie_d_cantidad": 8,
"especie_e_cantidad": 8 }

Rondas

GET /rondas/{id_partida} → Lista las rondas de una partida

POST /rondas → Crea una ronda (solo 2 rondas por partida)

Ejemplo: { "id_partida": 1, "numero_ronda": 1 }

Turnos

GET /turnos/{id_ronda} → Lista los turnos de una ronda

POST /turnos → Registra un turno (12 turnos por ronda)

Ejemplo: { "id_partida": 1, "id_ronda": 1, "numero_turno": 1, "asiento_jugador": 1,
"lanzador_dado": 1, "restriccion_dado": "BOSQUE_O_RIO", "dinosaurios_recibidos":
["T_REX", "ESPECIE_A", "ESPECIE_B", "ESPECIE_C", "ESPECIE_D", "ESPECIE_E"],
"dinosaurio_colocado": "ESPECIE_A" }

Puntajes finales

GET /puntajes/{id_partida} → Obtiene los puntajes finales

POST /puntajes → Registra los puntajes al finalizar la partida

Ejemplo: { "id_partida": 1, "asiento_jugador": 1, "puntaje_total": 25, "puntos_rio": 8,
"puntos_bonus_trex": 6, "contador_trex_desempate": 3 }

Cómo probar la API

1. Navegador web:
Solo para métodos GET. Ejemplo:
<http://localhost/juegoDraftosaurus/partidas>
2. Postman:
 - Método: POST
 - URL: <http://localhost/juegoDraftosaurus/partidas>
 - Headers: Content-Type: application/json
 - Body: JSON con los datos requeridos
3. Terminal (curl):
curl -X POST <http://localhost/juegoDraftosaurus/partidas> -H "Content-Type: application/json" -d '{"nombre_jugador1":"Ana","nombre_jugador2":"Luis"}'

Solución de problemas comunes

- Error 404: Verifique que el archivo .htaccess exista y que Apache esté configurado correctamente.
- Error de conexión a la base de datos: Revise el archivo config/database.php (usuario y contraseña).
- “Endpoint no encontrado”: Asegúrese de que los nombres de carpetas y archivos estén en minúsculas.
- “Class not found”: Confirme que los nombres en los require_once coincidan con los nombres reales de los archivos.

24. Análisis Costo-Beneficio/ Factibilidad Económica

El proyecto Draftosaurus es un desarrollo académico sin fines de lucro, realizado por 3 estudiantes del ITI (José Madruga, Fabricio Vanrell y Facundo Benítez). Aunque no se comercializa, para fines de análisis se evalúa como si fuera un producto real con comercialización

Costo de construcción:

Honorarios del equipo de desarrollo:

-Salario real programador junior \$35000

(https://www.glassdoor.com/Salaries/montevideo-programador-j%C3%BAnior-salary-SRCH_IL.0,10_IM1735_KO11,29.htm)

-3 programadores + 6 meses duración aprox proyecto = \$630000

-Costo: \$630000



Herramientas:

-VS Code, Git, GitHub, Figma, GIMP, Canva → todas gratuitas.

Costo: \$0

Capacitación del equipo de desarrollo:

-No se realizaron cursos formales, talleres ni se compró bibliografía específica.

-El aprendizaje fue autodidacta (documentación online, clases del curso).

Costo: \$0

Lugar de trabajo:

-Trabajo remoto desde casa.

-Trabajo desde la UTU

Costo: \$0

Hardware:

-Cada integrante usó su propia computadora.

-Aunque ya estaban compradas antes del proyecto, si consideramos su uso exclusivo para el desarrollo durante 6 meses, se podría estimar un costo de depreciación.

-Sin embargo, como no hubo gasto nuevo durante el proyecto, se considera:

Costo: \$0

Total costo construcción: \$630000

Costos de implementación:

-Manuales de usuario e instalación → desarrollados por el equipo sin costo.

-Capacitación de usuarios → no aplica.

-Instalación → local con XAMPP, sin gastos.

Total: \$0

Costos de operación y mantenimiento

-El sistema se ejecuta localmente, sin servidores, hosting ni soporte externo.

Total: \$0

Beneficios tangibles:

-Ahorro en materiales físicos (no se imprimen cartas ni tableros).

-Pero como el proyecto es 100% académico y no se usa en escenarios reales masivos, este ahorro es mínimo o nulo.

Valor tangible: \$0



Beneficios intangibles:

- Experiencia real en desarrollo web (PHP, HTML, CSS, JS).
- Aplicación de metodologías de Ingeniería de Software (COCOMO, Gantt, casos de uso, gestión de riesgos, etc.).
- Trabajo en equipo bajo paradigma descentralizado democrático.
- Producto funcional para portafolio profesional.

Valor intangible: Muy alto

Conclusión: Nuestro único beneficio es el producto y ese beneficio es mayor al costo por ende nos vale mucho la pena

Tabla de finanzas y Plan Financiero (Emprendedurismo)

Paso también el plan de marketing hecho en emprendedurismo que tuvimos que fingir comprar computadoras, alquiler de local, insumos, tuvimos que agregar nuevas formas de ingreso ya que quedábamos muy en negativo y el profe quería que quedara positivo

Ingresos proyectados a los próximos seis meses

Definimos nuestros ingresos proyectados a partir de la venta de productos propios y accesorios tecnológicos. Elegimos combinar ambos rubros porque consideramos que esto diversifica las fuentes de ingreso y reduce los riesgos ante posibles fluctuaciones del mercado. Estimamos un ingreso mensual de \$182.000 UYU, lo que representa un total de \$1.092.000 UYU a seis meses.

Detalle de ingresos mensuales:

- Draftosaurus (juego): 80 unidades \times \$200 = \$16.000
- Remeras: 30 \times \$750 = \$22.500
- Llaveros: 70 \times \$100 = \$7.000
- Cargadores: 35 \times \$800 = \$28.000
- Vidrio + carcasa (instalación): 35 \times \$800 = \$28.000
- Auriculares: 30 \times \$600 = \$18.000
- Joystick para PC: 25 \times \$2.500 = \$62.500
- Total, mensual: \$182.000 UYU
- Total, a 6 meses: \$1.092.000 UYU

Gastos operativos estimados a seis meses

Incluimos todos los costos fijos necesarios para mantener el funcionamiento estable de la empresa, como alquiler, servicios básicos, sueldos, mantenimiento y aportes patronales. Creemos que esta estimación nos permite proyectar con precisión el flujo de caja y garantizar la continuidad de las operaciones sin sobresaltos financieros. Calculamos un gasto mensual de \$155.600 UYU, equivalente a \$933.600 UYU en seis meses.

Detalle de gastos operativos mensuales:

Alquiler: \$16.000
Gastos comunes: \$2.000
UTE: \$3.500
OSE: \$600
Telefonía: \$3.000
Mantenimiento: \$800
Seguros: \$1.200
Sueldos (3 socios × \$30.000): \$90.000
Aportes patronales (30%): \$27.000
Préstamo Fundasol (USD 300 ≈ \$11.700): \$11.700
Otros: \$1.500

Total, mensual: \$155.600 UYU
Total, a 6 meses: \$933.600 UYU

Costos de producción estimados a seis meses

Detallamos el costo unitario de cada producto, basándonos en los precios actuales de materiales y proveedores. No se asigna costo al juego Draftosaurus, ya que es de desarrollo propio. El costo total estimado es de \$76.750 UYU mensuales, equivalente a \$460.500 UYU en seis meses.

Detalle de costos variables mensuales:

Remeras: $30 \times \$300 = \9.000
Llaveros: $70 \times \$50 = \3.500
Cargadores: $35 \times \$300 = \10.500
Vidrio + carcasa: $35 \times \$150 = \5.250
Auriculares: $30 \times \$200 = \6.000
Joystick: $25 \times \$1.700 = \42.500
Draftosaurus: \$0

Total, mensual: \$76.750 UYU
Total, a 6 meses: \$460.500 UYU



Gastos de marketing y ventas estimados a seis meses

Decidimos enfocar nuestra inversión en redes sociales (Instagram, Facebook y TikTok) y en un cartel con el logo de la empresa para reforzar la identidad visual del local. Consideramos esta inversión esencial para posicionar la marca y generar reconocimiento.

Detalle de gastos de marketing a 6 meses:

Publicidad en redes sociales: $\$3.000 \times 6 = \18.000

Cartel con logo: \$2.000

Total: \$20.000 UYU

Activos con los que comienza la empresa

Optamos por invertir en elementos indispensables para operar desde el inicio. Esta estructura inicial, valorada en \$401.000 UYU, garantiza un inicio sólido y autosuficiente.

Detalle de activos iniciales:

Computadoras (3): \$90.000

Mobiliario: \$45.000

Equipos de oficina: \$10.000

Inventario inicial: \$120.000

Señal de alquiler: \$16.000

Gastos de constitución: \$20.000

Efectivo en caja: \$100.000

Total, activos iniciales: \$401.000 UYU

costos variables						
	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6
Remeras	\$9.000,00	\$9.000,00	\$9.000,00	\$9.000,00	\$9.000,00	\$9.000,00
Llaveros	\$3.500,00	\$3.500,00	\$3.500,00	\$3.500,00	\$3.500,00	\$3.500,00
Cargadores	\$10.500,00	\$10.500,00	\$10.500,00	\$10.500,00	\$10.500,00	\$10.500,00
Vidrio + carcasa (instalado)	\$5.250,00	\$5.250,00	\$5.250,00	\$5.250,00	\$5.250,00	\$5.250,00
Auriculares	\$6.000,00	\$6.000,00	\$6.000,00	\$6.000,00	\$6.000,00	\$6.000,00
Joystick para PC	\$42.500,00	\$42.500,00	\$42.500,00	\$42.500,00	\$42.500,00	\$42.500,00
Costo de producción mensual:	\$76.750,00	\$76.750,00	\$76.750,00	\$76.750,00	\$76.750,00	\$76.750,00
Total a 6 meses:						\$460.500,00



Costos fijos						
	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6
Alquiler:	\$16.000,00	\$16.000,00	\$16.000,00	\$16.000,00	\$16.000,00	\$16.000,00
Gastos comunes:	\$2.000,00	\$2.000,00	\$2.000,00	\$2.000,00	\$2.000,00	\$2.000,00
UTE:	\$3.500,00	\$3.500,00	\$3.500,00	\$3.500,00	\$3.500,00	\$3.500,00
OSE:	\$600,00	\$600,00	\$600,00	\$600,00	\$600,00	\$600,00
Telefonía:	\$3.000,00	\$3.000,00	\$3.000,00	\$3.000,00	\$3.000,00	\$3.000,00
Mantenimiento:	\$800,00	\$800,00	\$800,00	\$800,00	\$800,00	\$800,00
Seguros:	\$1.200,00	\$1.200,00	\$1.200,00	\$1.200,00	\$1.200,00	\$1.200,00
Sueldos (3 socios x \$30.000):	\$90.000,00	\$90.000,00	\$90.000,00	\$90.000,00	\$90.000,00	\$90.000,00
Aportes patronales (30%):	\$27.000,00	\$27.000,00	\$27.000,00	\$27.000,00	\$27.000,00	\$27.000,00
Préstamo Fundasol (USD 300 ≈ \$11.700):		\$11.700,00	\$11.700,00	\$11.700,00	\$11.700,00	\$11.700,00
Publicidad en redes sociales (Instagram, Facebook, TikTok)	\$3.000,00	\$3.000,00	\$3.000,00	\$3.000,00	\$3.000,00	
Insumo:	\$2.000,00		\$2.000,00		\$2.000,00	
Otros:	\$1.500,00	\$1.500,00	\$1.500,00	\$1.500,00	\$1.500,00	\$1.500,00
Total mensual:	\$150.600,00	\$160.300,00	\$162.300,00	\$160.300,00	\$162.300,00	\$157.300,00
Total 6 meses						\$953.100,00

Inversión inicial	
rubro	\$
Computadoras	\$90.000,00
Mobiliario (mesas, sillas, mostrador):	\$45.000,00
Equipos de oficina (microondas, etc.)	\$10.000,00
Inventario inicial:	\$120.000,00
Señal de alquiler (1 mes):	\$16.000,00
DGI	\$6.550,00
BPS	\$5.913,00
Efectivo en caja:	\$100.000,00
Total activos iniciales:	\$393.463,00

Ingresos proyectados						
	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6
Draftosaurus juego	\$16.000,00	\$16.000,00	\$16.000,00	\$16.000,00	\$16.000,00	\$16.000,00
Remeras	\$22.500,00	\$22.500,00	\$22.500,00	\$22.500,00	\$22.500,00	\$22.500,00
Llaveros	\$7.000,00	\$7.000,00	\$7.000,00	\$7.000,00	\$7.000,00	\$7.000,00
Cargadores	\$28.000,00	\$28.000,00	\$28.000,00	\$28.000,00	\$28.000,00	\$28.000,00
Vidrio + carcasa (instalado)	\$28.000,00	\$28.000,00	\$28.000,00	\$28.000,00	\$28.000,00	\$28.000,00
Auriculares	\$18.000,00	\$18.000,00	\$18.000,00	\$18.000,00	\$18.000,00	\$18.000,00
Joystick para PC	\$62.500,00	\$62.500,00	\$62.500,00	\$62.500,00	\$62.500,00	\$62.500,00
Préstamo Fundasol (USD 10.000 = \$420.000):	\$420.000,00					
Ingreso mensual total:	\$602.000,00	\$182.000,00	\$182.000,00	\$182.000,00	\$182.000,00	\$182.000,00
Ingreso proyectado a 6 meses:						\$1.512.000,00



25. Tamaño punto por función

TIPO	ELEMENTO	DESCRIPCION
Entrada Externa (EE)	Registro de jugadores	El usuario ingresa nombre(s) → datos entran al sistema
EE	Selección de cartas en draft	El usuario elige cartas → el sistema actualiza estado
EE	Colocación de dinosaurios	El usuario arrastra fichas al tablero → se actualiza partida
EE	Guardar partida (si aplica)	El usuario guarda estado del juego
Consulta Externa (CE)	Ver reglas del juego	Muestra info sin cálculo
CE	Ver puntuación actual	Muestra puntos sin procesamiento complejo
Salida Externa (SE)	Mostrar resultado final	Calcula y muestra puntuación total (tiene lógica de cálculo)
Archivo Logico Interno (ALI)	Partida actual	Datos de jugadores, turnos, tablero, cartas → mantenidos en memoria o archivo
ALI	Configuración del juego	Idioma, modo (digital/seguimiento), etc.

Valor de dominio de información	Conteo		Factor ponderado				
			Simple	Promedio	Complejo		
Entradas externas (EE)	<input type="text"/>	×	3	4	6	=	<input type="text"/>
Salidas externas (SE)	<input type="text"/>	×	4	5	7	=	<input type="text"/>
Consultas externas (CE)	<input type="text"/>	×	3	4	6	=	<input type="text"/>
Archivos lógicos internos (ALI)	<input type="text"/>	×	7	10	15	=	<input type="text"/>
Archivos de interfaz externos (AIE)	<input type="text"/>	×	5	7	10	=	<input type="text"/>
Conteo total							<input type="text"/>

Factor ponderado (simple)

$$EE = 4 \times 3 = 12$$

$$CE = 2 \times 4 = 6$$

$$SE = 1 \times 4 = 4$$

$$ALI = 2 \times 7 = 14$$

$$\text{Conteo total} = 36$$

Factores de Ajuste de Valor (FAV)



Nº	Pregunta	Puntaje
1	Comunicación de datos	2
2	Procesamiento distribuido	0
3	Rendimiento crítico	1
4	Uso hardware existente	2
5	Transacciones	3
6	Entrada interactiva	4
7	Eficiencia	2
8	Actualización online	0
9	Complejidad de procesamiento	2
10	Reusabilidad	1
11	Facilidad de conversión/instalación	3
12	Facilidad de operación	4
13	Múltiples instalaciones	2
14	Facilidad de mantenimiento	3

Suma de FAV = 29

PF = Conteo total x (0.65 + (0.01 x FI)) = 36 x (0.65 + (0.01 x 29)) = 33.84

Php = 12.70 Trabajo en horas por PF

12.70 x 33.84 = 429768 hs

429768 / 8 = 53721 días laborales aprox

53721 / 5 = 10,7 semanas aprox

53721 / 20 = 2,7 meses aprox

2,7 / 3 = 1 mes por programador aprox

35000 (sueldo junior) x 3 = 105000

26. Analisis FODA ponderado

Fortalezas (internas)

ITEM	PESO	CLASIF	VALOR PONDERADO	
Somos 3 integrantes	0.12	4	0.48	
Alta creatividad	0.12	4	0.48	
Buena organización	0.14	3	0.42	
Colaboración mutua y apoyo constante entre integrantes.	0.14	4	0.56	
Uso efectivo de herramientas	0.10	3	0.30	
Capacidad de autogestión: dividimos tareas sin necesidad de un líder fijo.	0.18	4	0.72	
Motivación por entregar un producto funcional y presentable.	0.20	4	0.80	TOTAL= 3.76

Debilidades (internas)

ITEM	PESO	CLASIF	VALOR PONDERADO	
Tendencia a la procrastinación	0.14	1	0.14	
Falta de experiencia grande	0.12	2	0.24	
Tiempo limitado	0.15	1	0.15	
Conocimiento limitado	0.10	2	0.20	
Solo somos 3 integrantes	0.13	1	0.13	
Alta dependencia de cada uno.	0.12	2	0.12	
Falta de comunicación en momentos críticos	0.10	2	0.10	
Dificultad para estimar tiempos reales	0.14	1	0.14	TOTAL= 1.44

INTERNAS= 5.2



Oportunidades (externas)

ITEM	PESO	CLASIF	VALOR PONDERADO	
Uso de IA como apoyo	0.15	3	0.45	
Retroalimentación de los profesores.	0.10	4	0.40	
Contactos que pueden derivar a futuros puestos de trabajo.	0.15	4	0.60	
Alta demanda de programadores.	0.20	3	0.60	
Validación académica: si el proyecto destaca, puede usarse en futuras ediciones del curso.	0.20	4	0.80	
Innovación en ideas	0.20	4	0.80	TOTAL= 3.65

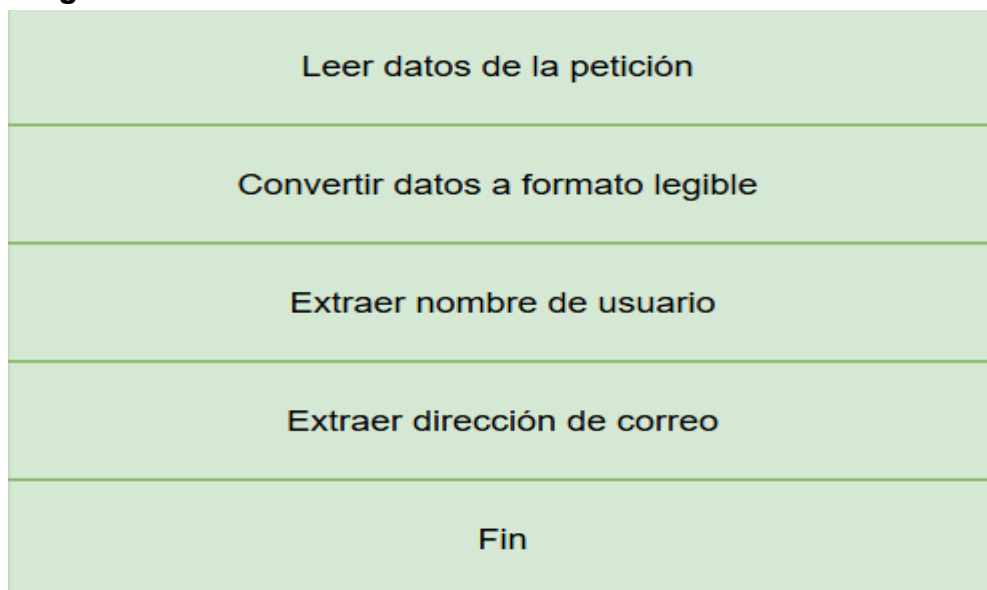
Amenazas (externas)

ITEM	PESO	CLASIF	VALOR PONDERADO	
Alta competencia	0.10	3	0.30	
Carga académica intensa	0.15	2	0.30	
Dependencia total del 100% de participación de cada integrante.	0.14	2	0.28	
Otros equipos están más avanzados o tienen mejor diseño.	0.09	3	0.27	
Riesgo de que el juego no funcione correctamente el día de la defensa (error técnico).	0.16	2	0.32	
Cambio de requisitos de último momento	0.12	3	0.36	
Evaluación subjetiva	0.24	2	0.48	TOTAL= 2.31

EXTERNAS = 5,96

27. Diagramas Nassi-Shneiderman (N-S)

Diagrama 1: Estructura Secuencial

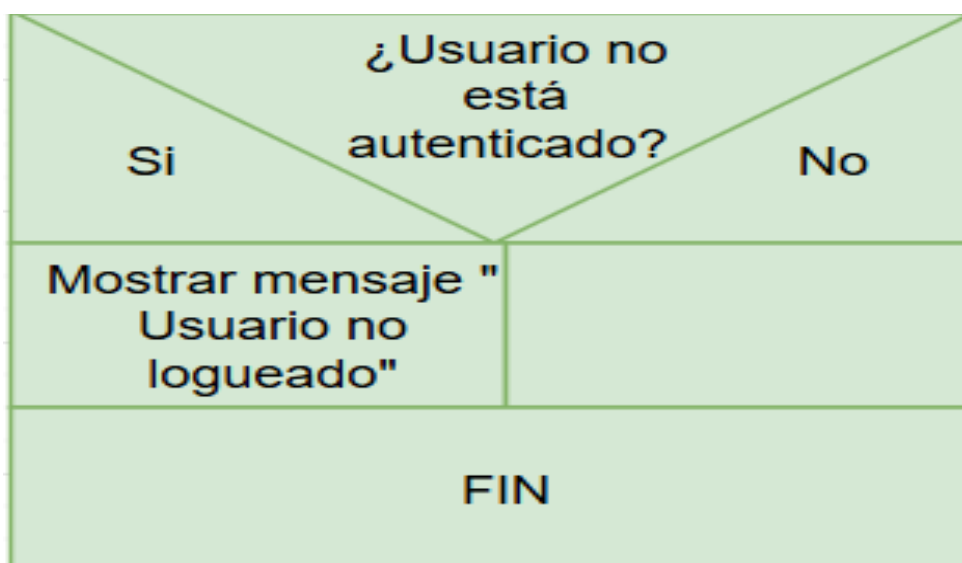


Fragmento de código fuente: AuthController.php

```
$raw = file_get_contents("php://input");
$data = json_decode($raw, true);

$username = isset($data['username']) ? trim((string)$data['username']) : '';
$email = isset($data['email']) ? trim((string)$data['email']) : '';
```

Diagrama 2: Estructura Selectiva Simple

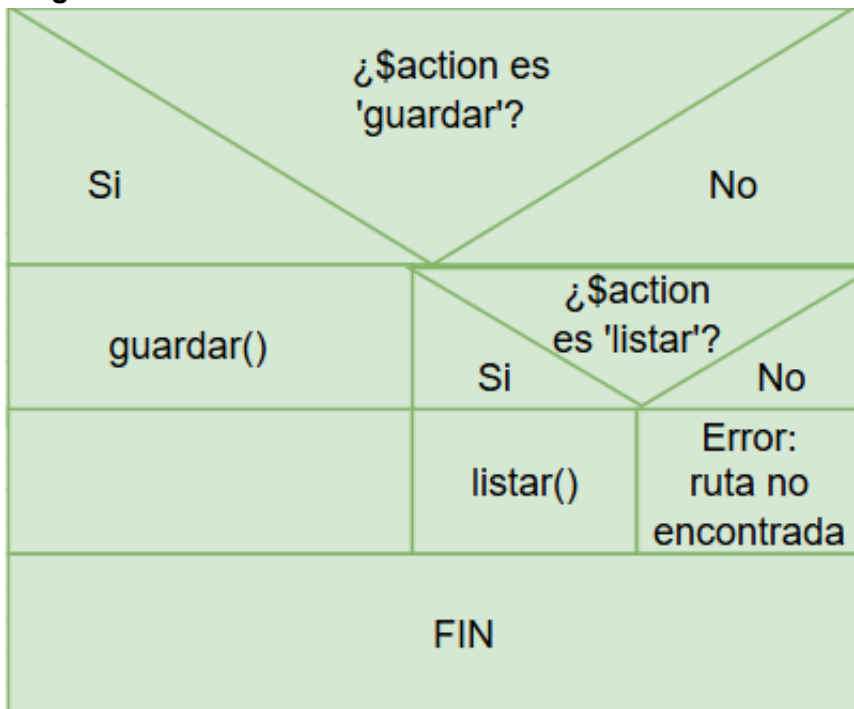


Fragmento de código fuente: PartidaController.php

```
$userId = $_SESSION['user_id'] ?? null;

if (!$userId) {
    echo json_encode(['success' => false, 'message' => 'Usuario
no logueado']);
    return;
}
```

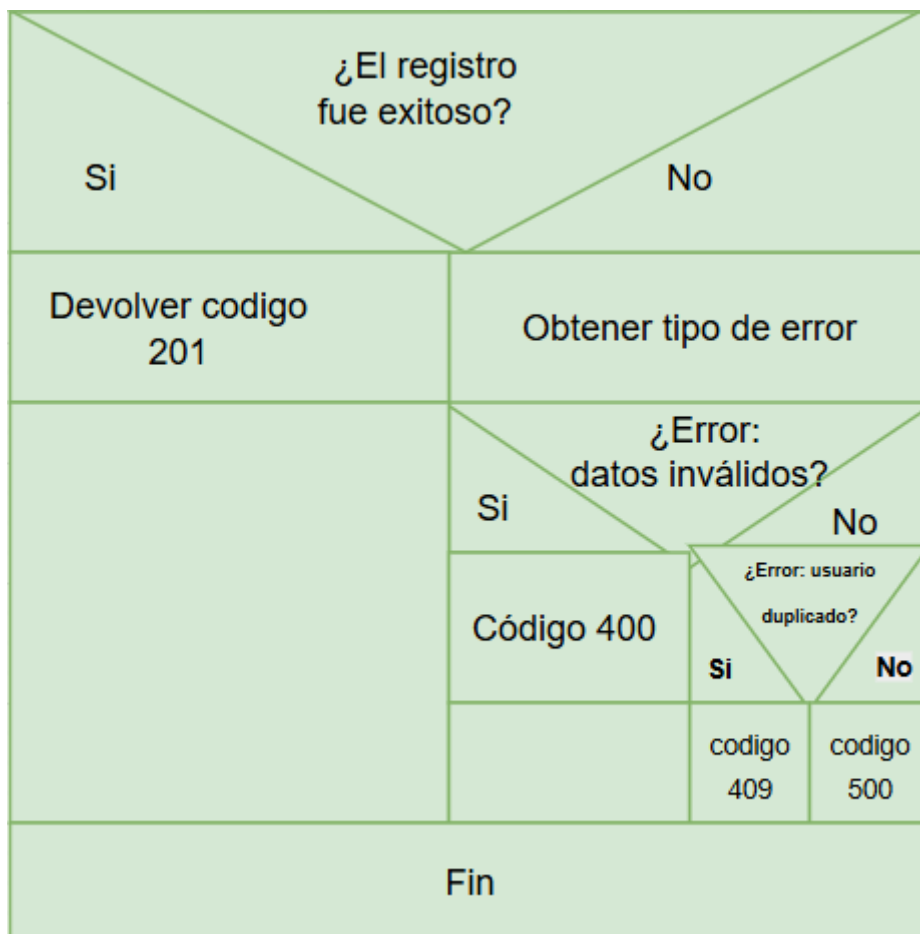
Diagrama 3: Estructura Selectiva anidada



Fragmento de código fuente: Partidas.php

```
if ($action === 'guardar') {
    $controller->guardar();
} elseif ($action === 'listar') {
    $controller->listar();
} else {
    echo json_encode(['success' => false, 'message' => 'Ruta no
encontrada']);
}
```

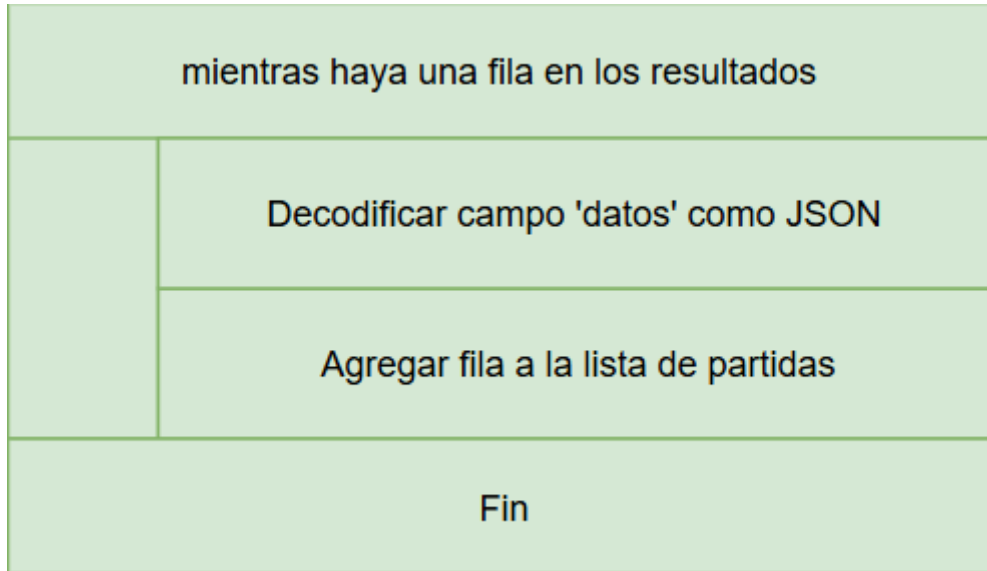
Diagrama 4: Estructura Selectiva Múltiple anidada



Fragmento de código fuente: AuthController.php

```
if ($result['success'] == true) {
    http_response_code(201);
} else {
    $code = isset($result['code']) ? $result['code'] : 'error';
    if ($code == 'invalid') {
        http_response_code(400);
    } elseif ($code == 'duplicate') {
        http_response_code(409);
    } else {
        http_response_code(500);
    }
}
```

Diagrama 5: Estructura Repetitiva



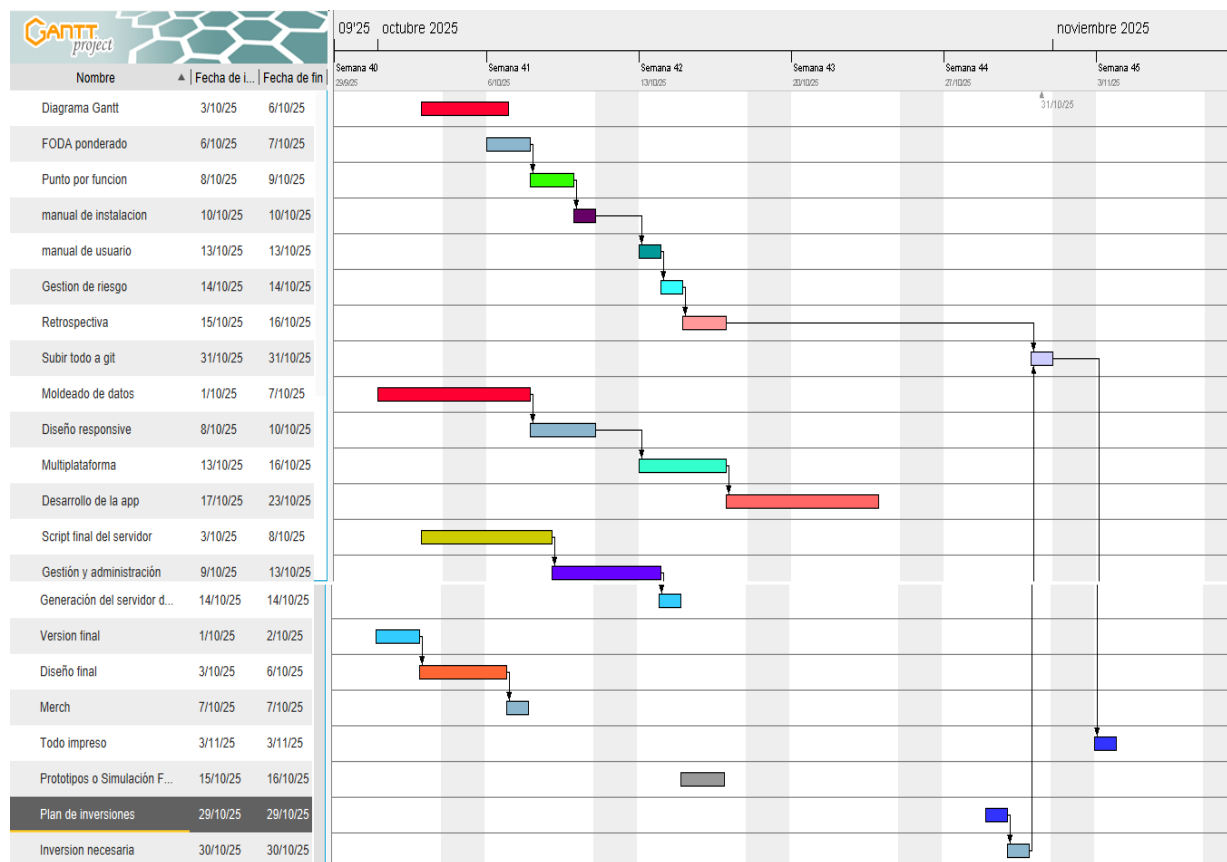
Fragmento de código fuente: AuthController.php

```
while ($row = $result->fetch_assoc()) {  
    $row['datos'] = json_decode($row['datos'], true);  
    $partidas[] = $row;  
}  
return $partidas;  
}
```

28. Diagrama de Gantt

La entrega inicio el 01/10/2025 finaliza el día 03/11/2025

(va adjunto el archivo del diagrama en GanntProyect)



29. Redacción retrospectiva

Cuando iniciamos este proyecto allá por mayo, lo hicimos sabiendo que se nos venía un último año muy cargado en todos los sentidos de la palabra y así fue. Fuimos atravesando etapas, clases y momentos hasta llegar hasta hoy noviembre último mes de clases a 2 semanas de defender este proyecto y egresar del ITI y aun no lo creemos y todo lo que tuvimos que atravesar, días calurosos, noches frías, discusiones, procrastinación, estrés, satisfacción, alegría, etc.

En todo este transcurso aprendimos muchas cosas: cómo armar una API desde cero, cómo organizar el código para que otro lo entienda, cómo lidiar con errores que parecían no tener solución... pero también aprendimos a lidiar con nosotros mismos. A manejar la presión de saber que esto contaba para la calificación final, pero que también era nuestra carta de presentación como futuros técnicos.

Aprendimos que fallar no es retroceder: es parte del proceso. Y que cada bug arreglado, cada reunión de última hora, cada cambio de requisitos, nos fue preparando no solo para entregar un proyecto, sino para enfrentar lo que venga después del ITI.

Para finalizar queremos quedarnos con todo lo bueno, que el esfuerzo constante rinde frutos, que confiar en tu equipo es tan importante como saber programar, y que a veces lo más difícil no es el código, sino sentarse a hacerlo cuando todo en la cabeza dice “déjalo para mañana”.

Hoy, a dos semanas de terminar, miramos atrás y no podemos creer que ya hayan pasado tantos meses y que ya seguramente no nos vamos a volver a ver mas con muchos de nuestros compañeros. No podemos creer que ya terminamos el proyecto después de tantos pensamientos negativos y creer que no llegábamos

Queremos agradecerle a todos los profes que nos acompañaron en este transcurso y agradecerles la constante presión ya que eso nos impulsa y después de todo es lo mejor para nosotros y nuestro futuro.

Gracias atentamente: Tricotech