

S.I.G.P.D.

Ingeniería de Software

CodeFlow

Rol	Apellido	Nombre	C.I	Email
Coordinador	Manzinalli	Felipe	5.714.301-9	felipe.manzinalli@estudiante.ceibal.edu.uy
Sub-Coordinador	Suarez	Pablo	5.656.677-9	psuarez@estudiante.ceibal.edu.uy
Integrante 1	Maldonado	Agustin	5.617.059-8	ezequiel.maldonado@estudiante.ceibal.edu.uy
Integrante 2	Diez	Daniel	5.547.065-6	danieldiez05@gmail.com

Docente: Pereira, Diego

**Fecha de
culminación**

14/7/2025

PRIMERA ENTREGA

ÍNDICE

Documentación de inicio y planificación de proyecto	2
Reglamento Interno de CodeFlow	2
Diagrama de Gantt para la primera entrega	7
Relevamiento de Datos	8
Documentos Utilizados	8
Clasificación de Aspectos Relevantes	8
Observación Directa	9
Encuesta	9
Estudio de factibilidades	15
1. Factibilidad Técnica	15
2. Factibilidad Económica	15
3. Factibilidad Operativa:	16
4. Factibilidad legal:	16
Especificación de requerimientos (IEEE 830)	17
1. Introducción	17
2. Descripción General	19
3. Requerimientos Específicos	21
Implementación del modelo de desarrollo	24
Nota aclaratoria	26
Repositorio de GitHub	26

Documentación de inicio y planificación de proyecto

Reglamento Interno de CodeFlow

Introducción:

El presente reglamento interno de CodeFlow tiene como objetivo establecer las normas y procedimientos para asegurar el buen funcionamiento del equipo, garantizar la productividad y mantener un ambiente de trabajo respetuoso y colaborativo. En este documento se detallan las formas de toma de decisiones, las normas sobre asistencia y participación en reuniones, el sistema de sanciones por incumplimiento, los medios de comunicación utilizados, así como los protocolos de apelación y posibles expulsiones en caso de conductas graves. El reglamento busca fomentar la organización, el respeto mutuo y el compromiso de cada miembro del grupo.

Sección I: Forma de Trabajo

Artículo 1 - Toma de decisiones

Las decisiones del grupo deberán tomarse de forma unánime, es decir, todos los miembros deben llegar a un acuerdo tras debatir el tema correspondiente.

Artículo 2 - Intervención del coordinador

Si no se logra un acuerdo entre todos los integrantes, el coordinador tomará una decisión desde una postura imparcial, teniendo en cuenta el contexto y el problema en cuestión.

Artículo 3 – Comunicación de inconvenientes urgentes

Cualquier inconveniente urgente relacionado con el avance del proyecto, que afecte directamente la tarea de un miembro o el desarrollo del mismo, debe ser comunicado inmediatamente al grupo. Si el miembro afectado lo considera necesario, podrá convocar una reunión informal para discutir y manejar la situación de manera oportuna y adecuada.

Artículo 4 – Información del proyecto

Toda información relevante del proyecto debe ser compartida con el equipo de forma transparente y oportuna.

Artículo 5 – Modificación del reglamento

Este reglamento puede ser modificado una vez aprobado. Para realizar cualquier cambio, se debe contar con el consentimiento unánime de todos los miembros.

Artículo 6 – Validez de las normas

El desconocimiento de una norma no exime a ningún miembro de su cumplimiento.

Sección II: Reuniones Formales**Artículo 7 – Reuniones formales**

Las reuniones formales se realizarán todos los jueves a las 17:30 en el ITI, extendiéndose hasta tratar todos los temas del orden del día. Al finalizar cada reunión formal, se deberá elaborar un acta detallada que refleje lo tratado, las decisiones tomadas y cualquier tema pendiente para la siguiente reunión.

Artículo 8 – Cambio de lugar

Si el lugar habitual no está disponible, se debe proponer con antelación un lugar alternativo y comunicarlo a todos los miembros. Si un integrante no responde al llamado de cambio y el resto acepta el nuevo lugar, el integrante que no respondió no asistirá y será sancionado.

Artículo 9 – Reunión híbrida (mitad presencial, mitad virtual).

En el caso de que un miembro no pueda asistir a una reunión formal de manera presencial, pero el resto del equipo esté realizando la reunión de forma presencial, el miembro ausente deberá unirse a la reunión de manera virtual (por medio de Discord u otro medio acordado). La reunión se llevará a cabo de manera híbrida (mitad presencial, mitad virtual), con la participación activa del miembro que no pudo asistir físicamente.

Si el miembro ausente se niega a unirse de manera virtual sin justificación válida, se le aplicará una sanción correspondiente por inasistencia a la reunión formal.

Artículo 10 – Adelanto de reunión

Si todos los integrantes tienen disponibilidad previa al horario pactado, la reunión podrá adelantarse, siempre que haya acuerdo unánime.

Artículo 11 – Reunión virtual obligatoria

Si tres o más miembros no pueden asistir a la reunión presencial, esta se convierte automáticamente en una reunión virtual obligatoria a la misma hora.

Artículo 12 – Temario

Cada reunión formal debe tener un temario a desarrollar.

Artículo 13 – Temas nuevos

Si surge un nuevo tema durante la reunión, se tratará en la siguiente sesión formal.

Sección III: Reuniones Informales

Artículo 14 – Reunión informal virtual

Las reuniones informales se realizarán 100% de forma virtual para tratar temas urgentes del proyecto o resolver inconvenientes inmediatos. Estas reuniones no requieren de una agenda predefinida, pero deben tener un propósito claro.

Artículo 15 - Cancelación de reunión informal

Si un miembro no se une a la reunión informal en la fecha y hora pactada, la reunión se cancelará y el miembro ausente será sancionado. Si la reunión informal es convocada

para más de 2 integrantes, la reunión no se cancela si al menos 2 miembros pueden asistir.

Artículo 16 – Acta de reunión informal

Cada reunión informal debe contar con un acta que resuma los acuerdos alcanzados. Este acta debe ser registrada en el canal de texto del servidor de Discord.

Sección IV: Sanciones

Serán pasibles de sanción los integrantes que incurran en alguna de las siguientes conductas:

- Agresión física o verbal (de cualquier tipo, en cualquier contexto).
- Incumplimiento de tareas asignadas que afecten el avance del proyecto.
- Ausencia sin aviso o justificación en reuniones formales o informales.
- Eliminación deliberada del trabajo del proyecto sin justificación.

Artículo 18 – Escala de sanciones

Las sanciones se aplicarán de manera acumulativa según la siguiente escala:

- 1 sanción: Se registra la sanción sin aplicar medidas inmediatas.
- 2 sanciones: Cambio de nombre en Instagram y WhatsApp por 24 horas, a elección del grupo.
- 3 sanciones: Cambio de foto de perfil en Instagram y WhatsApp por 24 horas, a elección del grupo.
- 4 sanciones: Asistencia con pijama de dinosaurio en la siguiente reunión formal.
- 5 sanciones: Grabar un audio explicando cómo sobrevivir a una invasión de dinosaurios (mínimo 30 segundos) y mandarlo al grupo de la clase por 24 horas.
- 6 sanciones: Grabar un audio cantando una canción seleccionada por el grupo (2-4 minutos) y mandarlo al grupo de proyecto.
- 7 sanciones: Asistir a clase con championes de diferentes colores o diseños (excepto en días de presentación o defensa).
- 8 sanciones: Comer una comida elegida por los demás integrantes.
- 9 sanciones: Realizar una broma telefónica a alguien, elegida por los integrantes del grupo.

- 10 sanciones: Permitir que los demás manipulen su celular durante 3 minutos.
- Más de 10 sanciones: El tiempo de manipulación del celular se incrementará en 3 minutos por cada sanción adicional.

Artículo 19 - Acumulación de sanciones

Las sanciones se limpiarán cada dos meses. Si un miembro acumula sanciones después de este período, empezará de nuevo con el contador en cero.

Artículo 20 – Sanciones por eliminación de trabajo

Si un miembro elimina todo o parte del trabajo del proyecto sin aviso previo ni justificación, se le sancionará inmediatamente con el uso de su teléfono, de acuerdo con la cantidad de sanciones acumuladas previamente.

Artículo 21 – Apelación de sanciones

Si un miembro considera que una sanción no fue justa o correctamente aplicada, podrá apelar. La apelación deberá ser aprobada por el resto del equipo, y si se determina que la sanción fue inapropiada, esta será retirada.

Sección V: Evaluación de Expulsión**Artículo 22 - Causas de expulsión**

Los siguientes comportamientos serán motivo de evaluación para expulsión del equipo:

- Violencia física grave o amenazas creíbles hacia otro miembro.
- Agresión verbal sistemática que cree un entorno hostil.
- Eliminación deliberada y masiva del trabajo del proyecto.
- Reiteradas ausencias y desinterés absoluto, cuando el equipo determine que no se puede contar con esa persona.
- Incumplimiento reiterado de normas de convivencia.
- Compartir información confidencial sin permiso del equipo.
- Acumulación excesiva de sanciones.

En caso de cumplirse todo los puntos, se evaluará la expulsión del miembro ante el cuerpo docente, que tomará la decisión final.

Sección VI: Medios de Comunicación

WhatsApp: Para mensajes rápidos, coordinación de reuniones informales y resolución de dudas.

Discord: Para reuniones formales y virtuales, registro de sanciones y actas.

Google Drive: Carpeta compartida con todos los documentos del proyecto, actas y recursos relevantes

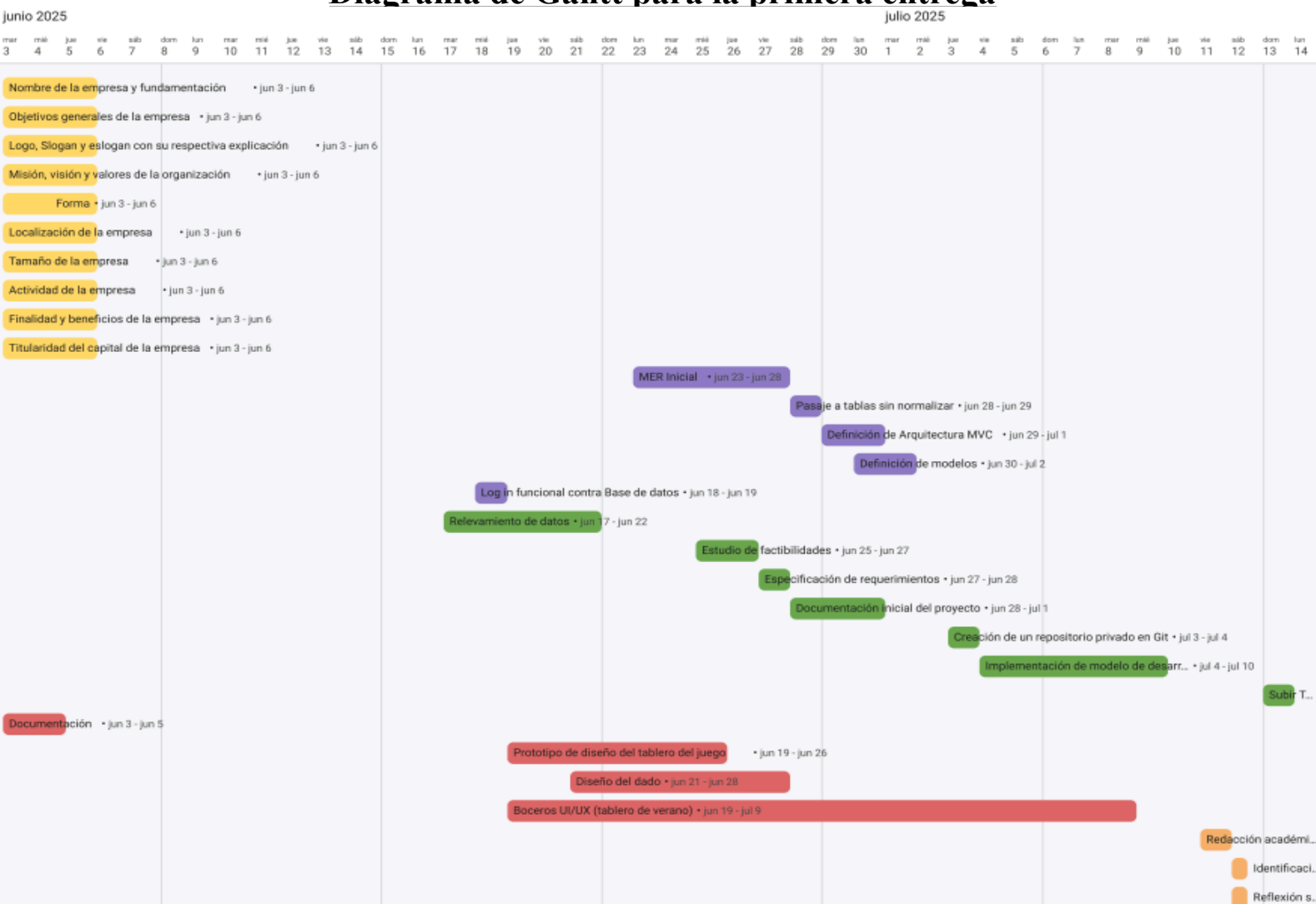
Trello: Gestión de tareas y seguimiento de progreso.

Figma: Herramienta para crear y compartir los diseños del proyecto.

GitHub: Repositorio de código y control de versiones.

Miro: Herramienta colaborativa para plasmar avances visuales y organizativos del proyecto

Diagrama de Gantt para la primera entrega



Relevamiento de Datos

Documentos Utilizados

- Letra del proyecto
- Reglas oficiales del juego Draftosaurus
- Video explicativo del juego
- Guía para Relevamiento de Datos

Clasificación de Aspectos Relevantes

Aspectos básicos:

- Aplicación web desarrollada en PHP, HTML, CSS y JavaScript.
- Registro manual de colocación de dinosaurios en el tablero
- Cálculo automático de puntajes por parte del sistema
- Validación de reglas y restricciones durante la partida
- Base de datos implementada en MySQL
- Estructura lógica de Backend y Frontend

Aspectos Esenciales:

- Implementación de modo de seguimiento de partida.
- Desarrollo de modo de juego digitalizado.
- Validación de las restricciones del dado y reglas de los recintos
- Sistema de perfiles para jugadores y administradores

Aspectos Secundarios:

- Integración con sistema de control de versiones GitHub

- Diseño responsivo adaptado para celulares y tablets

Aspectos Ornamentales:

- Estética visual coherente con la experiencia física del juego
- Navegación simple y clara
- Animaciones e interacciones agradables para el usuario
- Mensajes de error y validación contextualizados

Observación Directa**Descripción:**

Se jugó una partida del juego físico Draftosaurus. Participaron todos los integrantes del grupo en una partida completa de dos rondas.

Tipo de observación: Participativa estructurada.

Duración: Aproximadamente 25 minutos (una partida completa)

Registro: Se anotaron observaciones sobre dinámicas, errores comunes, ritmo de juego y experiencia de usuario.

Principales hallazgos y conclusiones:

- Las reglas de colocación de dinosaurios en los recintos y las restricciones del dado fueron elementos centrales del juego y generaron varios errores de interpretación
- Se detectó una necesidad recurrente de revisar el reglamento durante el juego, especialmente al puntuar.
- La dinámica de paso de dinosaurios y el dado de colocación agregan valor estratégico y deben ser replicadas con precisión.
- El flujo del juego es ágil, por lo que la interfaz digital debe ser intuitiva, rápida y amigable, sin entorpecer la experiencia grupal.
- Se concluyó que una herramienta digital puede aumentar significativamente la fluidez del juego, al automatizar puntuaciones y validar reglas en tiempo real.

1. Público Objetivo

El público objetivo de esta encuesta fueron personas de diversas edades, con o sin experiencia en juegos de mesa, que potencialmente podrían ser usuarios de una aplicación web auxiliar para gestionar partidas del juego Draftosaurus. Se priorizó la heterogeneidad en cuanto a edades y hábitos digitales para evaluar la viabilidad del proyecto y comprender las necesidades de una posible audiencia real.

2. Objetivo de la Encuesta

El objetivo principal fue relevar la percepción, experiencia previa, necesidades y expectativas de potenciales usuarios respecto a una aplicación que sirva como apoyo para contar puntos, validar reglas y mejorar la experiencia de juego de Draftosaurus. Se buscó validar la utilidad percibida, identificar dispositivos preferidos y funcionalidades prioritarias.

3. Diseño del cuestionario

El cuestionario estuvo compuesto por 10 preguntas cerradas, diseñadas para obtener información sobre los siguientes ejes:

- Perfil demográfico (edad)
- Experiencia en juegos de mesa
- Frecuencia y dispositivo de uso de apps
- Percepción de utilidad de la aplicación
- Características esperadas
- Preferencias de interfaz y funcionalidades adicionales

El formato incluyó preguntas de selección única y múltiple, para permitir mayor riqueza en las respuestas.

4. Estrategia de Entrega

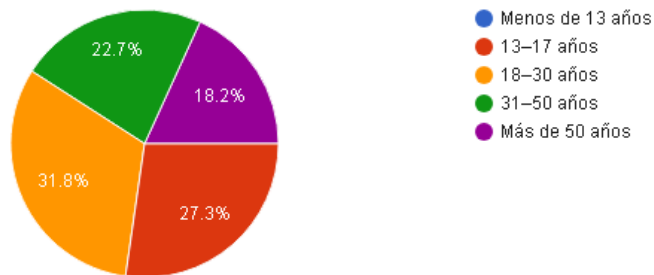
La encuesta fue distribuida en formato digital. Se aseguraron criterios de anonimato y voluntariedad. Las preguntas fueron claras, breves y adaptadas a distintos niveles de familiaridad con la tecnología. Participaron 22 personas que completaron la encuesta en su totalidad.

5. Procesamiento de Datos

¿Qué edad tenés?

22 responses

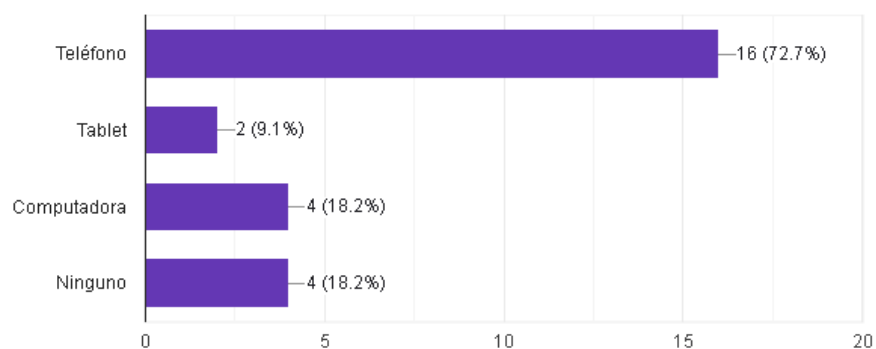
 Copy chart



¿Con qué dispositivos usas más aplicaciones?

22 responses

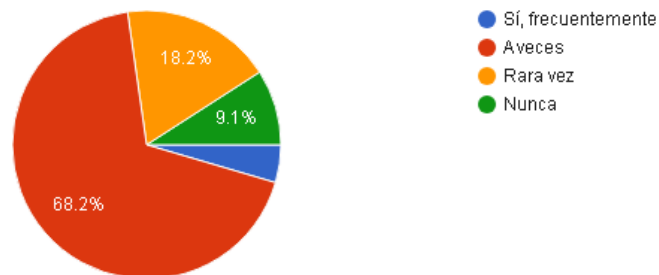
 Copy chart



¿Has jugado alguna vez juegos de mesa?

 Copy chart

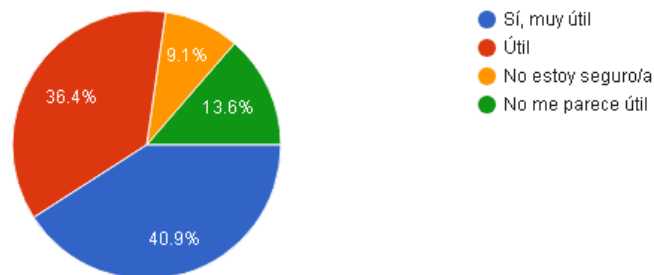
22 responses



¿Te parece útil una aplicación que ayude a contar puntos y validar reglas en un juego de mesa?

 Copy chart

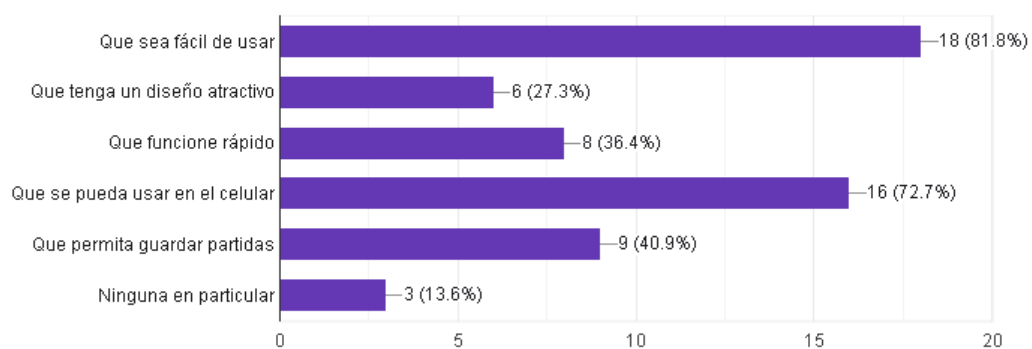
22 responses



¿Qué características crees que serían más importantes en una aplicación de este tipo?

 Copy chart

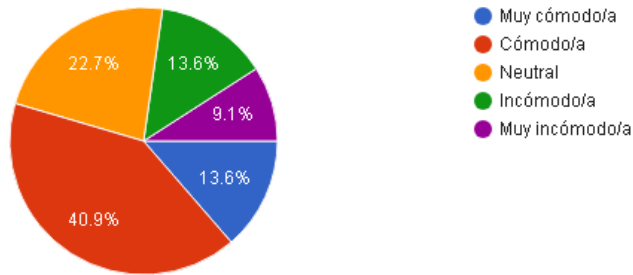
22 responses



¿Qué tan cómodo/a te sientes usando aplicaciones web?

 [Copy chart](#)

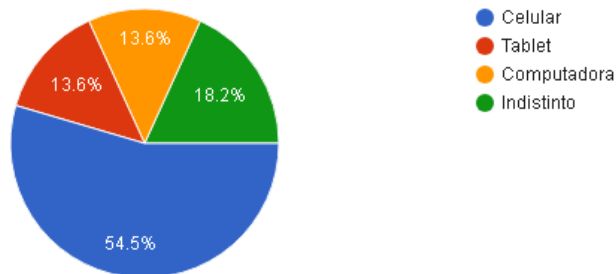
22 responses



Si tuvieras que usar una aplicación para ayudar en un juego de mesa, preferirías usarla en:

 [Copy chart](#)

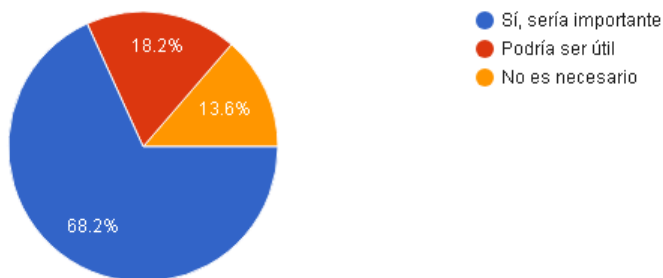
22 responses



¿Te gustaría que la aplicación incluya un tutorial que explique cómo jugar el juego de mesa?

 [Copy chart](#)

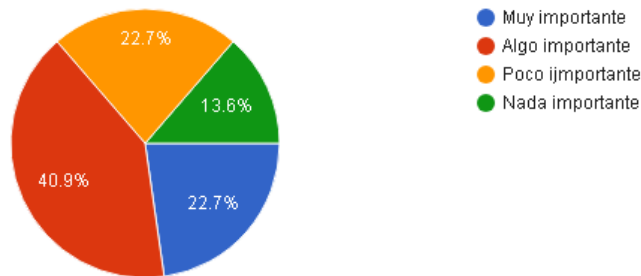
22 responses



¿Qué tanto te interesa que la aplicación tenga animaciones y sonidos?

 Copy chart

22 responses



Si tuvieras que elegir, ¿qué prefieres?

 Copy chart

22 responses



6. Conclusiones generales

El cuestionario evidencia que:

- Existe interés real y validación de la utilidad de una aplicación auxiliar para juegos de mesa.
- Los usuarios valoran la simplicidad, portabilidad y guía interactiva.
- El desarrollo debe centrarse en la versión móvil, con interfaz clara, rápida y accesible.
- Se recomienda incorporar funciones de enseñanza junto con el contador de puntos, para aprovechar el deseo del usuario por aprender jugando.
- Las funcionalidades ornamentales (sonido, animaciones) deben ser secundarias, pero podrían estar habilitadas como opción personalizable.

Estudio de factibilidades

1. Factibilidad Técnica

Relevamiento de necesidades:

- Desarrollo de una aplicación web en PHP, JavaScript, HTML y CSS.
- Base de datos MySQL para almacenar partidas y configuraciones
- Diseño responsivo para celulares y computadoras
- Implementación de lógica del juego (turno, puntuación, validación de reglas).
- Validación en tiempo real de reglas del juego.

Recursos Disponibles:

- Conocimientos de HTML, CSS, JavaScript y PHP básicos.
- Experiencia inicial con MySQL.
- Herramientas: XAMPP, Visual Studio Code, GitHub.
- Integrantes del equipo con computadoras propias para trabajar desde casa.
- Manuales y documentación del juego.
- Software Open source: PHP, MySQL, Apache, Visual Studio Code, GitHub.

Recursos futuramente Disponibles:

- Servidor GNU/Linux para despliegue final
- Optimización de consultas SQL.

Evaluación:

Factibilidad técnica viable con la siguiente condición:

- El equipo debe reforzar conocimientos en: HTML, CSS, JavaScript y PHP.

2. Factibilidad Económica

Relevamiento de necesidades:

- Licencias de software gratuitas, al ser herramientas open-source.
- Sueldo de cada integrante del equipo de desarrollo.
- Simulación de una inversión empresarial.
- Costos asociados a desarrollo, despliegue y mantenimiento.

Recursos Disponibles:

- Capital simulado aportado por los socios: inversión inicial de los integrantes de CodeFlow, destinada a cubrir los costos principales de desarrollo, infraestructura y operación inicial.
- Préstamo simulado complementario: recurso financiero adicional que permitirá solventar gastos asociados al despliegue y mantenimiento.
- Espacio físico de trabajo (oficina) y gastos comunes, (internet, energía eléctrica, agua).

Recursos futuramente disponibles:

- Fondo de contingencia para cubrir gastos imprevistos relacionados con infraestructura o soportes técnicos de la aplicación.

Evaluación:

Económicamente viable porque, aunque simula un escenario empresarial con inversiones y préstamos, los costos reales son mínimos al utilizar herramientas open-source (PHP, MySQL, HTML, CSS) y recursos institucionales gratuitos.

3. Factibilidad Operativa:

Relevamiento de necesidades:

- El proyecto debe integrarse con asignaturas del curso
- Los usuarios finales deben encontrar la herramienta intuitiva

Recursos Disponibles:

- Equipo humano: 4 integrantes con roles definidos
- Organización de los tiempos: mediante diagramas de Gantt y modelos de desarrollo.

4. Factibilidad legal:

Relevamiento de necesidades:

- Uso de propiedad intelectual: El sistema utiliza directamente la marca Draftosaurus y reproduce el juego en su totalidad de manera digital.. Esto sí infringe derechos de autor y propiedad intelectual de los creadores del juego.

Recursos futuramente disponibles:

- Se debería obtener una autorización expresa de los titulares de la propiedad intelectual del juego.

Evaluación:

viable con la siguiente condición:

- En este contexto, se justifica el uso mediante el principio de “uso educativo no comercial”. Sin embargo, en un contexto real, este proyecto no podría ser realizado sin autorización legal..

Especificación de requerimientos (IEEE 830)

1. Introducción

1.1 Propósito

Este documento define los requerimientos del sistema I.G.P.D., una aplicación web que permite jugar al juego de mesa Draftosaurus en modo digital o usarla como apoyo en partidas físicas. Está dirigido a docentes, desarrolladores, clientes, testers y futuros mantenedores.

1.2 Ámbito del Sistema

El sistema a desarrollar, denominado IGPD, será una aplicación web diseñada para asistir y/o replicar partidas del juego de mesa Draftosaurus en su modo base.

S.I.G.P.D

ITI

3MD

El sistema contará con dos modos principales de funcionamiento:

- **Modo Seguimiento:** permitirá a los jugadores registrar manualmente la colocación de dinosaurios en sus parques durante una partida física. El sistema validará las reglas del juego y calculará automáticamente la puntuación final, agilizando el proceso de control y conteo de puntos.
- **Modo Juego Digitalizado:** ofrecerá una versión completamente digital del juego, replicando su mecánica original. Incluirá la extracción aleatoria de dinosaurios desde una reserva virtual, la selección de uno por turno, el paso de los dinosaurios restantes al siguiente jugador, la colocación de los dinosaurios en los recintos según las reglas del juego y la validación automática de las restricciones impuestas por el dado. El sistema también calculará y mostrará los puntajes al final de la partida.

Este sistema no sustituye el juego físico, pero busca complementar la experiencia de juego mediante herramientas digitales que mejoren la gestión, control y disfrute de las partidas.

1.3 Definiciones, acrónimos y abreviaturas

- **Backend:** Parte lógica del sistema que procesa datos, ejecuta reglas del negocio y se comunica con la base de datos. En este proyecto se desarrolla en PHP.
- **CRUD** (*Create, Read, Update, Delete*): Operaciones básicas sobre bases de datos.
- **CSS** (*Cascading Style Sheets*): Hojas de estilo para presentación visual de la interfaz.
- **Frontend:** Parte visible del sistema con la que interactúa el usuario directamente. Incluye la UX/UI, desarrollada con HTML, CSS y JavaScript.
- **HTML** (*HyperText Markup Language*): Lenguaje de marcado para estructuras web.
- **IGPD:** Sistema Informático de Gestión de Partidas para Draftosaurus.
- **ITI:** Instituto Tecnológico de Informática.

- **JS** (*JavaScript*): Lenguaje de programación del lado del cliente, utilizado para la interacción dinámica del sistema.
- **LAMP**: Conjunto de tecnologías compuesto por Linux, Apache, MySQL/MariaDB y PHP, utilizado como entorno de ejecución del sistema.
- **PHP** (*Hypertext Preprocessor*): Lenguaje de programación del lado del servidor.
- **RF**: Requerimiento Funcional.
- **RNF**: Requerimiento No Funcional.
- **UI** (*User Interface*): Interfaz visual a través de la cual el usuario interactúa con el sistema.
- **UX** (*User Experience*): Experiencia global del usuario al interactuar con el sistema, enfocada en usabilidad y accesibilidad.
- **XAMPP**: Paquete de software que permite ejecutar un entorno LAMP de manera local en sistemas operativos Windows, Linux y MacOS.

1.4 Referencias

- Letra del Proyecto ITI 2025
- Manual de Draftosaurus (versión PDF)
- Norma IEEE 830

2. Descripción General

2.1 Perspectiva del producto

I.G.P.D. es una aplicación web diseñada como complemento y adaptación digital del juego de mesa Draftosaurus. El juego consiste en construir un parque temático de dinosaurios a través de una mecánica de drafting y colocación estratégica de dinosaurios en recintos con reglas específicas.

Draftosaurus se juega en dos rondas de seis turnos cada una. En cada turno, los jugadores seleccionan un dinosaurio al azar, lanzan un dado que impone una restricción de colocación, eligen uno de sus dinosaurios para colocarlo en su parque y luego pasan los restantes al siguiente jugador. Cada recinto del parque tiene condiciones particulares

de puntuación. Al final del juego, se suman los puntos en función de la colocación de los dinosaurios.

El sistema I.G.P.D. replica o asiste este proceso en formato digital a través de dos modos de uso:

- **Modo Seguimiento:** permite a los jugadores registrar su progreso en una partida física, aplicando reglas y sumando puntos automáticamente.
- **Modo de Juego Digitalizado:** emula completamente la mecánica del juego, incluyendo la selección aleatoria de dinosaurios, el paso entre jugadores, las restricciones del dado, la validación de reglas y el conteo automático de puntos.

El producto es autónomo, pero se apoya en tecnologías web y puede ejecutarse en un servidor LAMP. Además, busca mantener la experiencia original del juego físico, ofreciendo a su vez ventajas de accesibilidad, automatización y registro de partidas.

2.2 Funciones del Producto

- Registro y gestión de partidas (modo seguimiento y digital)
- Validación automática de reglas
- Cálculo y visualización de puntajes
- Visualizar mano de dinosaurios, seleccionar uno para colocarlo y transferir automáticamente los dinosaurios restantes al siguiente jugador.
- Gestión de usuarios y roles (jugador/administrador)

2.3 Características de los Usuarios

El sistema I.G.P.D. estará destinado principalmente a dos tipos de usuarios:

Jugadores:

- **Perfil general:** Niños, adolescentes y adultos que disfrutan de juegos de mesa familiares. El juego Draftosaurus está recomendado para mayores de 8 años, por lo que la interfaz deberá ser accesible y comprensible para usuarios desde esa edad en adelante.
- **Edad estimada:** entre 8 y 60 años.

- Nivel técnico: En su mayoría, usuarios no técnicos, con experiencia mínima o nula en entornos informáticos avanzados.

Administradores del sistema:

- Perfil general: Usuarios con conocimientos técnicos encargados del mantenimiento, configuración y gestión del sistema.
- Edad estimada: entre 20 y 50 años, usualmente estudiantes avanzados o egresados del área de la informática
- Nivel técnico: Medio a alto. Deberán ser capaces de trabajar con servidores LAMP, bases de datos MySQL/MariaDB y código fuente en PHP y JavaScript.

2.4 Restricciones

- Lenguaje backend: PHP 8.x
- Lenguaje frontend: JavaScript, HTML, CSS3
- Base de datos: MySQL
- Hosting: Entorno LAMP
- Aplicación bilingüe: Español/Inglés

2.5 Suposiciones y Dependencias

- Se asume que el usuario tiene acceso a un navegador moderno.
- El sistema correrá en infraestructura establecida por la institución.
- Se usará GitHub como plataforma de versionado.

3. Requerimientos Específicos

3.1 Requerimientos Funcionales

Alta prioridad:

RF01. Registro de jugadores

- El sistema debe permitir registrar jugadores con nombre y correo electrónico

RF02. Ingreso de dinosaurios

S.I.G.P.D

ITI

3MD

- El jugador podrá seleccionar dinosaurios y colocarlos en recintos específicos.

RF03. Validación de restricciones por dado y reglas de colocación

- El sistema debe aplicar automáticamente las reglas impuestas por los dados y los recintos

RF04. Cálculo automático de puntaje final

- El sistema debe calcular el puntaje total de cada jugador al finalizar la partida.

RF05. Juego digitalizado por completo

- El sistema debe permitir jugar partidas digitales de Draftosaurus, incluyendo sorteo de mano de dinosaurios, paso y colocación de dinosaurios.

Prioridad media:

RF06. Lanzamiento del dado con restricción

- El sistema debe simular el dado de colocación e imponer su restricción a los demás jugadores.

RF07: Asignación automática de dinosaurios por ronda

- Al inicio de cada ronda, el sistema debe asignar aleatoriamente una mano de 6 dinosaurios a cada jugador.

RF08: Paso de dinosaurios entre jugadores

- El sistema debe transferir automáticamente los dinosaurios no elegidos al siguiente jugador en orden.

RF09. Gestión de rondas y turnos

- El sistema debe llevar un control automático de turnos y rondas, finalizando la partida tras la segunda ronda.

RF10. Colocación en el Río

- El sistema debe permitir colocar un dinosaurio en el Río si no puede ser colocado en un recinto válido.

RF11. Detección de colocaciones inválidas

- El sistema debe impedir la colocación de dinosaurios en recintos que violen sus reglas.

RF12. Visualización del tablero por el jugador

- Cada jugador debe ver claramente su propio tablero con los recintos y dinosaurios colocados.

RF13. Finalización automática de partida

- El sistema debe detectar cuándo se han completado todas las rondas y mostrar el resultado final.

RF14. Reseña final con desglose de puntos

- Al finalizar la partida, el sistema debe mostrar un resumen por recinto, puntos extra y total.

RF15. Aplicación de criterios de desempate

- En caso de empate, el sistema debe aplicar los criterios oficiales

RF16. Aplicación de criterios de desempate

- El sistema debe permitir a un usuario con rol administrador gestionar partidas, configuraciones y mantenimiento general.

3.2 Requerimientos No Funcionales

Alta prioridad:

RNF1: UX/UI intuitiva y clara

- El diseño de la experiencia e interfaz debe ser simple, accesible y apto para usuarios desde 8 años.

RNF2. Disponibilidad en entorno local (LAMP)

- El sistema debe funcionar correctamente en servidores locales usando Apache, PHP y MySQL.

RNF3. Seguridad básica (acceso restringido)

- El acceso al panel de administración debe estar protegido por autenticación

Prioridad media:

RNF4.. Rendimiento aceptable

- Todas las acciones clave deben ejecutarse en menos de 10 segundos

RNF5. Portabilidad y diseño responsivo

- La aplicación debe ser usable en computadoras, tablets y smartphones con diseño adaptable.

Prioridad baja:**RNF6. Mantenibilidad del sistema**

- El código debe estar estructurado, comentado y documentado para facilitar futuras modificaciones.

RNF8. Accesibilidad visual

- El diseño debe utilizar íconos claros, contraste adecuado y evitar barreras visuales para usuarios con dificultades

3.3 Alcance**Incluye:**

- Seguimiento y puntuación de partidas físicas
- Juego digital completo
- Validación de reglas y lógica del juego
- Gestión de usuarios y partidas

Excluye:

- Juego en red o multijugador online
- Integración con redes sociales
- Inteligencia artificial (juego contra la máquina)

3.4 Limitaciones

- No se puede jugar online.
- La lógica de la IA para jugadores automáticos no está implementada
- No se cuenta con detección automática de errores humanos
- Compatible únicamente con navegadores modernos.

Implementación del modelo de desarrollo

Consideramos que para este proyecto es adecuado el uso del **modelo de desarrollo en cascada** debido a los siguientes puntos:

1. Requisitos estables y detallados desde el inicio
 - El alcance funcional (modo seguimiento y digital), restricciones técnicas (PHP, MySQL) y criterios de logro están bien definidos.
 - No se espera que surjan cambios radicales de alcance.
 - El Modelo en Cascada es idóneo cuando los requisitos no evolucionan significativamente.
2. Necesidad de entregas claras y controladas
 - El proceso del proyecto exige múltiples entregas obligatorias con rúbricas específicas.
 - Cada fase (análisis, diseño, implementación, pruebas) debe documentarse exhaustivamente.
 - El modelo en cascada permite organizar y validar cada entrega de forma ordenada.
3. Facilita la evaluación y el seguimiento docente
 - Permite a los docentes verificar de manera secuencial cada etapa.
 - La rigidez del modelo reduce la ambigüedad sobre qué se evalúa en cada fase.
 - La documentación completa que genera es un insumo obligatorio del curso.
4. Escasa necesidad de interacción continua con usuarios finales
 - Aunque hay validación, no se requiere interacción iterativa constante.
 - El producto final se verá principalmente en la instalación y demostración.
5. Menor complejidad técnica y regulatoria
 - El sistema no requiere la gestión de cambios continuos como en un proyecto ágil.
 - No hay integración con sistemas externos que cambien en tiempo real.

Por lo tanto consideramos que el modelo en cascada es la opción más coherente porque se adapta bien a requisitos claros y definidos, asegura documentación detallada para evaluar entregas, facilita la planificación y control del proyecto por fases y reduce riesgos de desorganización en un contexto educativo con regulación y fechas fijas.

Nota aclaratoria

Las tareas correspondientes a la asignatura de Sistemas Operativos no se encuentran incluidas en el diagrama de Gantt ni fueron realizadas, debido a la ausencia prolongada de la docente responsable por motivos de salud (infección respiratoria).

Repositorio de GitHub

Se deja a continuación el enlace al repositorio de GitHub correspondiente a la primera entrega:

<https://github.com/codeflow838/CodeFlow.git>