S.I.G.P.D.

Ingeniería de software

JurassiCode

Rol	Apellido	Nombre	C.I	Email
Coordinador	Fianza	Ignacio	5.690.153-1	businessignaciofianza@gmail.com
Sub-Coordinador	Benítez	Sebastián	5.652.044-4	sebastianbenitez2505@gmail.com
Integrante 1	Fleitas	Joaquín	5.570.982-3	joacolambru7@gmail.com
Integrante 2	Paz	Tomás	5.700.344-1	tomaslautaropaz@gmail.com

Docente: Cairús, Brandon

Fecha de culminación

14/7/2025

PRIMERA ENTREGA

ÍNDICE

Ingeniería de software

REPOSITORIO DE GitHub:

https://github.com/JurassiCode/PROYECTO

Relevamiento aplicado al Juego Draftosaurus

1. Introducción

Durante una clase práctica, se utilizó el juego de mesa *Draftosaurus* como una dinámica lúdica con el fin de aplicar técnicas de relevamiento de requerimientos en un entorno simulado. A pesar de que el juego no constituye un sistema informático, su análisis permitió ejercitar habilidades esenciales para la comprensión, identificación y documentación de requerimientos funcionales y no funcionales, como se realiza en proyectos reales de Ingeniería de Software.

2. Resumen del Juego

Draftosaurus es un juego de mesa basado en la mecánica de selección por “draft”, en el cual cada jugador debe construir un parque jurásico personal seleccionando dinosaurios. Las decisiones están condicionadas por reglas específicas de colocación, y un dado introduce restricciones aleatorias en cada ronda. El objetivo del juego consiste en obtener la mayor cantidad de puntos al finalizar la partida, dependiendo de la distribución estratégica de los dinosaurios en las diferentes zonas del tablero individual.

3. Aplicación de Técnicas de Relevamiento

A continuación, se describen las técnicas de relevamiento aplicadas durante la experiencia y su correspondencia con prácticas reales en el desarrollo de software.

3.1 Cuestionarios (Implícitos)

Si bien no se utilizaron cuestionarios estructurados o escritos, se formularon preguntas concretas entre los jugadores antes y durante la partida. Estas consultas permitieron aclarar reglas y validar el entendimiento compartido del sistema.

Estas preguntas espontáneas ayudaron a identificar ambigüedades y divergencias en la interpretación de las reglas por parte de los usuarios.

Ejemplos:

- “¿Se puede repetir dinosaurio en esta zona?”
- “¿Cuándo se usa el dado?”

Este tipo de interacción simula el uso de cuestionarios abiertos y cerrados aplicados a usuarios reales para comprender funcionalidades, necesidades y expectativas en entornos de desarrollo.

3.2 Entrevistas (Informales)

Se llevaron a cabo conversaciones entre los participantes, especialmente con aquellos que ya conocían el juego. Estas entrevistas informales fueron fundamentales para comprender reglas complejas o mal interpretadas, así como para recoger experiencias previas.

Ejemplos:

- “¿Cómo jugaste vos la otra vez?”
- “¿Qué pasa si no me queda un lugar válido para poner el dinosaurio?”

Este tipo de diálogo reproduce las entrevistas realizadas en el contexto de proyectos informáticos, donde se consulta a usuarios o expertos en el dominio para clarificar requerimientos.

3.3 Etnografía (Observación Participativa)

Durante la partida, se observó el comportamiento de los jugadores, permitiendo identificar patrones, decisiones intuitivas y errores frecuentes. La observación participativa permitió comprender cómo interactúan los usuarios con un sistema, en este caso el juego, desde una perspectiva interna.

Ejemplos:

- Algunos jugadores evitaban consultar el manual y se guiaban por la experiencia de otros.
- Otros priorizaban ciertos tipos de dinosaurios de acuerdo con estrategias personales.

Esta técnica es equivalente a la etnografía en Ingeniería de Software, donde se observa el uso real de un sistema sin interferir en la actividad de los usuarios, a fin de comprender su comportamiento auténtico.

3.4 Análisis Bibliográfico (Manual del Juego)

Se consultó el manual oficial del juego para corroborar reglas y comportamientos observados durante la partida. El análisis documental es una técnica esencial para adquirir información precisa y estructurada sobre un sistema ya existente.

Ejemplos:

- Verificación de las condiciones para colocar dinosaurios en determinadas zonas.
- Confirmación del uso del dado y sus efectos en el desarrollo del juego.

Este análisis refleja la práctica común en el desarrollo de software de recurrir a documentación técnica o manuales de referencia para complementar y validar la información obtenida de otras fuentes.

4. Conclusión

La dinámica realizada permitió aplicar, en un entorno lúdico, diversas técnicas de relevamiento vistas en clase: cuestionarios, entrevistas, observación etnográfica y análisis bibliográfico. Aunque el sistema analizado no era informático, su estructura y reglas definidas facilitaron una simulación útil y representativa del trabajo de relevamiento en proyectos reales. Esta experiencia no solo contribuyó a comprender el juego, sino que también sirvió como ejercicio práctico para el desarrollo de competencias clave en Ingeniería de Software.

Estudio de Factibilidad

Proyecto: Software de Gestión de Partidas para Draftosaurus

1. Objetivo del Proyecto

El presente proyecto tiene como finalidad desarrollar una aplicación web que funcione como asistente digital para el juego de mesa *Draftosaurus*. Esta herramienta permitirá a los jugadores registrar sus jugadas, calcular los puntos de forma automática, llevar el control de las rondas, visualizar reglas específicas según la partida y almacenar el historial de sesiones jugadas. El sistema está diseñado para mejorar la experiencia de juego, minimizar errores manuales y ofrecer estadísticas que enriquezcan las partidas posteriores.

2. Tipos de Factibilidad

2.1 Factibilidad Técnica

Viabilidad: Alta

Desde el punto de vista técnico, el proyecto es completamente viable. Se utilizarán tecnologías accesibles y acordes a los conocimientos desarrollados en el curso:

- HTML y CSS: para la estructuración y estilización de la interfaz del sistema.
- Bootstrap: para asegurar un diseño responsivo, profesional y adaptable a diferentes dispositivos.
- PHP: para implementar la lógica del backend, incluyendo validaciones, reglas del juego y gestión de sesiones.
- MySQL: como sistema de almacenamiento de datos de jugadores, partidas y estadísticas.

- XAMPP: como entorno local de desarrollo que integra Apache y MySQL.

Ventajas Técnicas:

- Todas las tecnologías mencionadas son de código abierto y cuentan con amplia documentación.
- El equipo de trabajo ya ha utilizado o está familiarizándose con estas herramientas.
- Es posible construir los módulos esenciales (jugadores, partidas, puntuaciones) sin necesidad de tecnologías externas adicionales.

Desafíos Potenciales:

- Estructurar de forma eficiente la base de datos y la lógica del juego.
- Validar reglas dinámicamente según la condición del dado en cada ronda.
- Mantener la fluidez de la interfaz durante el desarrollo del juego.

Recomendaciones Técnicas:

- Dividir el desarrollo en módulos claramente definidos (jugadores, puntuación, reglas, historial).
- Utilizar Bootstrap para lograr una interfaz clara y simple.
- Realizar pruebas desde el comienzo en entorno local con XAMPP.

2.2 Factibilidad Económica

Viabilidad: Muy Alta

Este proyecto no implica costos económicos significativos para el equipo de desarrollo.

Costos Estimados:

Licencias: No se requiere inversión en licencias, dado que todas las tecnologías utilizadas son gratuitas.

Infraestructura: El desarrollo puede mantenerse en entorno local o utilizar alguna plataforma gratuita de alojamiento si se desea publicación.

Oportunidades Futuras:

- El sistema podría evolucionar hacia una herramienta en línea de acceso público.
- Existe potencial para ofrecer el software como recurso libre para la comunidad, siempre y cuando se respeten los aspectos legales correspondientes.

2.3 Factibilidad Operativa

Viabilidad: Alta

El sistema cumple una función clara y complementaria al juego físico. No intenta reemplazarlo, sino asistir a los jugadores en el desarrollo de la partida.

Beneficios Operativos:

- Registro automatizado de puntos y jugadas.
- Aplicación correcta y uniforme de las reglas del juego.
- Generación de estadísticas y/o rankings para jugadores frecuentes.

Riesgos Identificados:

- Algunos jugadores pueden resistirse al uso de tecnología durante un juego de mesa tradicional.
- Una interfaz mal diseñada o poco fluida puede entorpecer el ritmo del juego.

Soluciones Propuestas:

- Diseñar una interfaz simple, ligera y optimizada para su uso en dispositivos móviles.
- Desarrollar una versión mínima viable (MVP) enfocada en las funcionalidades esenciales: cálculo de puntos y control de rondas.

2.4 Factibilidad Legal y Ética

Viabilidad: Moderada

El juego Draftosaurus es una propiedad intelectual registrada, por lo que su uso implica ciertas restricciones legales.

Riesgos Legales:

- El uso del nombre del juego o material gráfico oficial puede infringir derechos de autor.
- No se debe presentar el software como herramienta oficial sin el correspondiente permiso.

Medidas Preventivas:

- Utilizar nombres alternativos como “Asistente de Partidas de Dinosaurios”.
- Evitar el uso de imágenes, íconos o contenido oficial del juego.
- Incluir un descargo de responsabilidad que indique el carácter no oficial, educativo y sin fines comerciales del proyecto.

2.5 Factibilidad de Tiempo

Viabilidad: Alta

El desarrollo del proyecto es factible dentro del plazo del semestre si se gestiona de forma organizada.

Plan Tentativo de Trabajo:

Etapa	*Duración Estimada*
Análisis y diseño inicial	***1 a 2 semanas***
Maquetado e interfaz (HTML/CSS/Bootstrap)	***2 a 3 semanas***
Programación de lógica con PHP	***2 a 3 semanas***
Integración con base de datos (MySQL)	***1 a 2 semanas***
Pruebas, ajustes y validaciones	***1 a 2 semanas***
Documentación y presentación final	***1 semana***

La planificación contempla márgenes de ajuste ante imprevistos, asegurando una entrega a tiempo con una versión funcional del sistema.

Actas de reunión:
https://drive.google.com/drive/folders/1I2-vKdz0Ktuk7k-5ynO1J2Hk5SaBG8dX?usp=sharing

Diagrama de gantt (descargar de drive e importar en https://www.onlinegantt.com/)
https://drive.google.com/drive/folders/1UJDQHNPj8VvL3SPgjLr2CrDLkEZjSdUw?usp=drive_link

Definición de Roles de Usuario, Permisos y Privilegios

Rol: Administrador

Descripción:

Usuario con acceso completo al sistema. Encargado de gestionar a otros usuarios, controlar las partidas y supervisar el funcionamiento general de la plataforma.

Permisos y privilegios:

- Crear, editar y eliminar usuarios (jugadores y otros administradores).
- Ver listado completo de usuarios registrados.
- Crear, editar y eliminar partidas.
- Asignar jugadores a partidas.
- Ver estadísticas de todas las partidas.
- Acceder al panel de administración completo.
- Ver y modificar configuraciones generales del sistema.
- No participa como jugador en una partida.

Rol: Jugador

Descripción:

Usuario final que participa en partidas del juego Draftosaurus. Tiene acceso limitado al sistema, centrado en su experiencia de juego.

Permisos y privilegios:

- Registrarse y autenticarse en el sistema.
- Ver su propio perfil y modificar datos personales (nombre, usuario, contraseña).
- Ingresar a partidas a las que fue asignado.
- Cargar sus movimientos o jugadas dentro de una partida.
- Ver resultados de partidas en las que participó.
- No puede crear, editar ni eliminar partidas.
- No puede ver ni administrar otros usuarios.

Especificación de Requerimientos (Funcionales, No Funcionales, Alcance y Limitaciones).

Alcance del sistema

El sistema a desarrollar es una aplicación web para la gestión de partidas del juego Draftosaurus, orientada al uso académico y recreativo. Permitirá a los usuarios registrarse como jugadores o administradores, gestionar partidas y registrar jugadas en cada turno. A su vez, los administradores podrán controlar tanto los usuarios como las partidas del sistema.

Requerimientos funcionales

RF1: El sistema debe permitir el registro de nuevos jugadores.
RF2: El sistema debe permitir el inicio de sesión con validación de credenciales.
RF3: El sistema debe distinguir entre jugadores y administradores al iniciar sesión.
RF4: El administrador debe poder crear, editar y eliminar usuarios.
RF5: El administrador debe poder crear, editar y eliminar partidas.
RF6: El administrador debe poder asignar jugadores a partidas.
RF7: El administrador debe poder ver estadísticas generales del sistema.
RF8: El jugador debe poder ver y editar su perfil.
RF9: El jugador debe poder ver las partidas en las que participa.
RF10: El jugador debe poder ingresar a partidas activas y registrar sus jugadas.
RF11: El jugador debe poder ver los resultados de sus partidas.

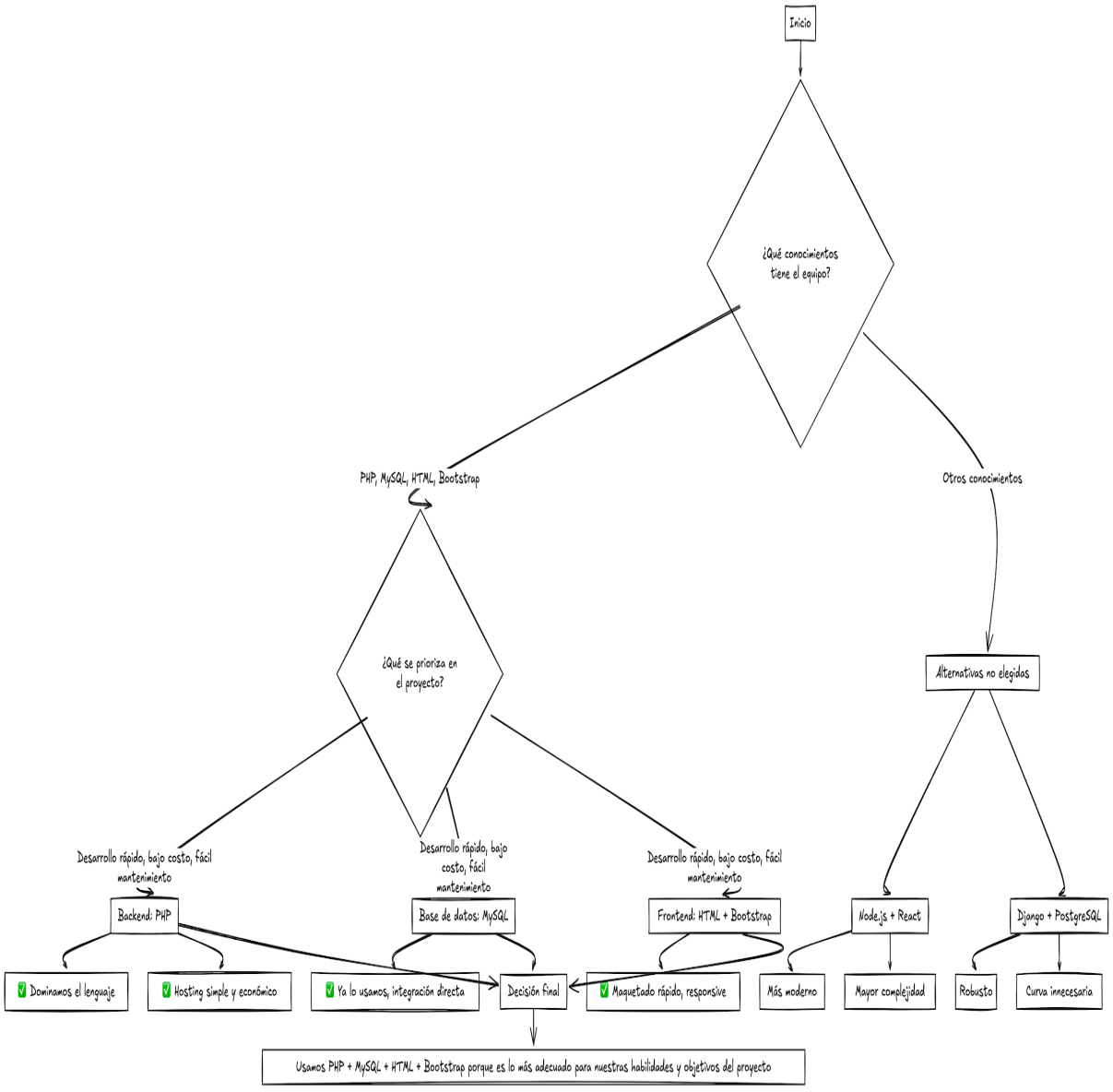
Requerimientos no funcionales

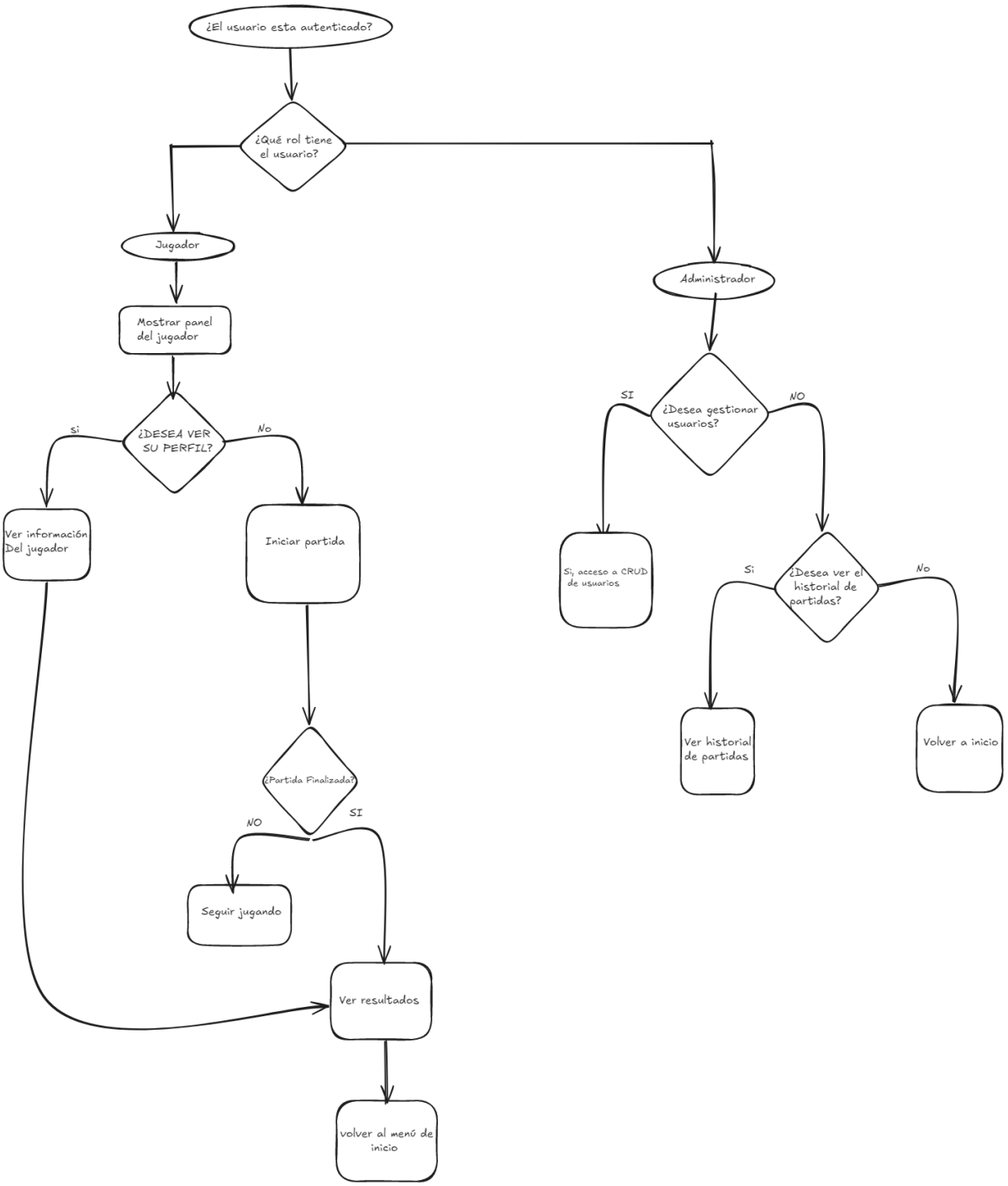
RNF1: El sistema debe cargar completamente en menos de 3 segundos.
RNF2: La interfaz debe ser responsive y adaptarse a dispositivos móviles y de escritorio.
RNF3: Las contraseñas deben almacenarse utilizando algoritmos de hashing seguro.
RNF4: El acceso a funciones debe estar restringido según el rol del usuario.
RNF5: Las sesiones deben expirar tras un período de inactividad.
RNF6: El sistema debe desarrollarse de manera modular para facilitar el mantenimiento.

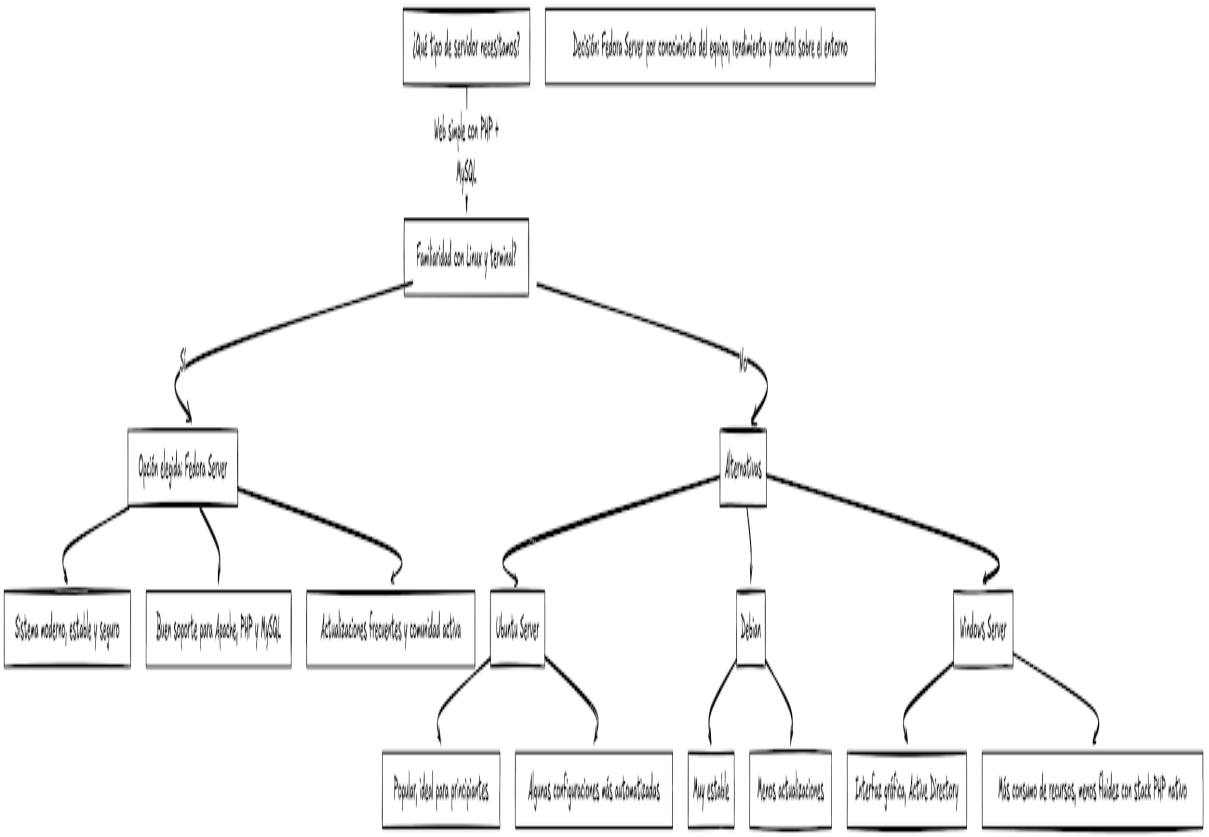
RNF7: La base de datos debe estar normalizada hasta la tercera forma normal (3FN).

RNF8: El sistema debe ser compatible con navegadores modernos (Chrome, Firefox, Edge).

Lógica del sistema y árboles de decisión




¿El usuario esta autenticado?
¿Qué rol tiene el usuario?
Jugador
Mostrar panel del jugador
¿DESEA VER SU PERFIL?
Si
No
Ver información Del jugador
Iniciar partida
¿Partida Finalizada?
NO
SI
Seguir jugando
Ver resultados
volver al menú de inicio
Administrador
¿Desea gestionar usuarios?
SI
NO
Si, acceso a CRUD de usuarios
¿Desea ver el historial de partidas?
Si
No
Ver historial de partidas
Volver a inicio



En caso de que las imágenes no se vean correctamente, dejamos acceso a la carpeta en drive de las mismas: https://drive.google.com/drive/folders/1qgAxLnHvzQJKm7OZR2NriBfwGfRdbp4S?usp=sharing

Ciclo de Vida Tradicional – Modelo en Cascada aplicado al proyecto

Idea central

El desarrollo del sistema se realiza paso a paso, sin retroceder. Cada etapa debe completarse totalmente antes de comenzar la siguiente.

1. Análisis

Objetivo:
Relevar y documentar todos los requisitos del sistema desde el inicio. En esta etapa se define qué debe hacer el sistema, sin pensar todavía en cómo lo va a hacer.

Aplicación al proyecto Draftosaurus:
En nuestro caso, lo primero fue entender el juego de mesa Draftosaurus para poder adaptarlo a una versión digital. Analizamos sus reglas, mecánicas y objetivos. A su vez, consultamos a docentes para conocer cómo pensaban usar el sistema con los estudiantes.

De esta forma, establecimos los **requisitos funcionales**, como:

- Registro y login de usuarios.
- Asignación de roles (jugador o administrador).
- Creación y administración de partidas.
- Registro de turnos y puntajes por jugador.

También definimos **requisitos no funcionales**, como:

- Buena velocidad de carga.
- Diseño intuitivo.
  Compatibilidad con dispositivos escolares.
- Seguridad básica de acceso.

Entregables de esta etapa:

- Documento con requisitos funcionales y no funcionales.
  Lista de funcionalidades que debe tener el sistema.
- Alcance y objetivos claros.

2. Diseño

Objetivo:
Planificar cómo se va a construir el sistema: tanto la parte visual como la estructura técnica. Se decide cómo se van a organizar las pantallas, formularios, funciones y base de datos.

Aplicación al proyecto:
En esta etapa pensamos cómo se iba a ver y comportar el sistema. Diseñamos los elementos principales de la interfaz (como formularios de registro, botones de acción y pantallas de juego), y los organizamos en bocetos simples o wireframes.

Además, planificamos cómo se iban a conectar las diferentes partes: la interfaz del usuario con la lógica del sistema y con la base de datos. También se definieron los permisos y funciones para cada tipo de usuario.

Entregables de esta etapa:

- Bocetos de cada pantalla del sistema.
- Plan general de organización del código.
- Definición de la estructura de base de datos.
- Documento técnico con decisiones de diseño.

3. Implementación

Objetivo:
Codificar todo el sistema siguiendo el diseño previamente planificado. Acá se transforma lo pensado en realidad.

Aplicación al proyecto:
Comenzamos escribiendo el código de las funciones básicas, como el registro, el inicio de sesión y el acceso a las distintas partes del sistema según el tipo de usuario. Se programó la lógica para administrar las partidas, registrar los turnos y almacenar los puntos de cada jugador.

Todo el desarrollo se hizo utilizando herramientas conocidas por el equipo, como PHP para la parte lógica, MySQL para la base de datos, y HTML/CSS/JS para la interfaz. El trabajo se organizó en módulos para mantener el código más claro y ordenado.

Entregables de esta etapa:

- Código completo del sistema.
- Sitio funcional en entorno local.
- Base de datos creada y conectada correctamente.

4. Testing

Objetivo:
Probar el sistema ya terminado para verificar que funcione como se espera. Se busca encontrar errores y asegurar que se cumplen los requisitos definidos en el análisis.

Aplicación al proyecto:
Realizamos pruebas con distintos tipos de usuarios, simulando situaciones reales de uso. Probamos todos los formularios, los botones, la lógica del turno por jugador y el sistema de puntajes. También hicimos pruebas en distintos dispositivos para asegurarnos de que la interfaz sea entendible y se vea bien.

Los errores encontrados fueron corregidos antes de dar por finalizado el sistema.

Entregables de esta etapa:

- Informe con resultados de las pruebas.
- Lista de errores corregidos.
- Sistema estable, listo para presentar.

Documentación de Inicio y Planificación del Proyecto

Nombre del Proyecto
JurassiDraft (Asistente Web para Partidas de Draftosaurus)

Integrantes del Grupo

- Ignacio Fianza (Coordinador)
- Sebastian Benitez (Sub Coordinador)
- Tomás Paz (Integrante 1)
- Joaquín Fleitas (Integrante 2)

Objetivo General
Desarrollar una aplicación web que sirva como asistente digital para el juego de mesa Draftosaurus, permitiendo registrar jugadores, jugadas, calcular puntos automáticamente y llevar el control de partidas, mejorando así la experiencia y la precisión durante el juego.

Alcance del Proyecto
- Registro de jugadores y creación de partidas.
- Carga de jugadas y rondas según las reglas del juego.
- Cálculo automático de puntajes.

- Visualización de resultados y estadísticas simples.
- Posibilidad de consultar el historial de partidas.
- Interfaz responsiva, usable desde celulares o computadoras.

Tecnologías Utilizadas

- Frontend: HTML, CSS, Bootstrap
- Backend: PHP
- Base de datos: MySQL
- Entorno de desarrollo: XAMPP (Apache + MySQL local)
- Organización del grupo: Hoja de cálculo compartida (Google Sheets)
- Comunicación: WhatsApp, Discord y reuniones presenciales en la UTU
- Control de Versiones: Git

Reglas del Grupo

1. Comunicación directa y continua
   Se mantiene contacto regular por WhatsApp o en persona durante horarios en la UTU. Cualquier duda o avance importante se comunica de inmediato.

2. Organización basada en tareas claras
   Las tareas están registradas en una hoja de cálculo compartida, con responsables asignados. Todos los integrantes pueden ver y actualizar el estado de las tareas.

3. Compromiso con el trabajo
   Cada integrante debe cumplir con las tareas asignadas en los tiempos acordados. En caso de imprevistos, se debe avisar con antelación al grupo.

4. Participación equitativa
   Todos colaboran tanto en el desarrollo como en el análisis y documentación del proyecto. Las decisiones importantes se discuten en conjunto.

5. Respeto y trabajo en equipo
   Se fomenta un clima de respeto mutuo, diálogo abierto y apoyo entre compañeros. Las diferencias se resuelven conversando.

6. Revisión antes de integrar
   Cada componente debe ser probado en local (XAMPP) antes de integrarse al proyecto general. El código debe estar validado para asegurar compatibilidad.

7. Gestión de versiones básica
   Aunque no se utiliza un sistema formal como Git, se conservarán copias funcionales del proyecto en cada avance importante para evitar pérdida de información.