



Root Group

~ The Root Group Company ~

# S.I.G.P.D.

(Programación Full Stack)

*The Root Group Company*

Rol	Apellido	Nombre	C.I	Email
Coordinador	Moreira	Luciano	5.664.037-3	luciano.moreira.uy@gmail.com
Sub-Coordinador	Ruiz	Gerónimo	5.699.972-8	geronimoruiz769@gmail.com
Integrante 1	Moreira	José	5.693.593-4	mateosignazio@gmail.com
Integrante 2	Lamarca	Santiago	5.600.318-3	santiagolamarca31@gmail.com

**Docente: Romero, Carlos**

**Fecha de culminación**

**14/07/2025**

**PRIMERA ENTREGA**

**I.S.B.O. ~ 3°MH**

*2025 © The Root Group Company*



Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

## Índice

<b>Análisis de PHP.....</b>	<b>3</b>
<b>Justificación de la elección de PHP.....</b>	<b>4</b>
<b>Selección de SGBD (MySQL/MariaDB).....</b>	<b>5</b>
<b>Justificación de la elección de MySQL.....</b>	<b>6</b>
<b>DIAGRAMA-ENTIDAD-RELACIÓN (D.E.R).....</b>	<b>7</b>
<b>RESTRICCIONES NO ESTRUCTURALES (R.N.E).....</b>	<b>8</b>
<b>Pasaje a Tabla.....</b>	<b>13</b>
<b>Formato de convención de Commits.....</b>	<b>15</b>
<b>Especificación de versiones.....</b>	<b>16</b>
<b>Fundamento de Git y GitHub.....</b>	<b>18</b>
<b>Fundamentación de Visual Studio Code.....</b>	<b>21</b>
<b>Justificación Front-end.....</b>	<b>24</b>
<b>Hoja Testigo.....</b>	<b>27</b>



Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

## **Análisis de PHP** **(ventajas ante Python o Node.js)**

### **Introducción:**

Este documento presenta un análisis comparativo entre PHP, Node.js y Python, enfocándose en las ventajas de PHP para el desarrollo web. El objetivo es fundamentar la elección de PHP como tecnología principal para el proyecto Draftosaurus.

En el desarrollo web existen múltiples opciones tecnológicas. Elegir la más adecuada impacta en la eficiencia y facilidad de desarrollo. Es necesario evaluar PHP frente a alternativas modernas como Node.js y Python para determinar cuáles ventajas tiene PHP.



Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

Característica	PHP	Node.js	Python
Facilidad de uso	Sintaxis sencilla, fácil integración con HTML	Requiere manejo de asincronía y callbacks	Sintaxis clara, pero más generalista
Hosting y costos	Compatible con hosting compartido económico	Requiere VPS o servidores dedicados	Similar a Node.js, depende del entorno
Comunidad y ecosistema	Amplia, con muchos CMS y frameworks	Comunidad activa, ecosistema npm	Comunidad grande, uso más allá del web
Rendimiento en Web Bueno en tareas	tradicionales y bases SQL	Excelente para aplicaciones en tiempo real	Más lento en ejecución, no compilado
Seguridad Funciones	integradas para sanitización	Seguridad depende de implementación	Seguridad depende de implementación
Base de Datos	Soporte robusto para bases relacionales	Mejor para bases NoSQL y tiempo real	Soporte variado, menos especializado en web



Root Group

~ *The Root Group Company* ~

Montevideo, 19 de mayo 2025

## **Justificación de la elección de PHP**

- **Facilidad y rapidez:** PHP permite un desarrollo rápido con menor curva de aprendizaje, ideal para estudiantes, caso del proyecto.
- **Costo y accesibilidad:** Amplio soporte en hosting económico, facilitando la implementación y despliegue.
- **Maleabilidad de desarrollo:** Gran comunidad y cantidad de frameworks y CMS que facilitan el desarrollo.
- **Integración a Bases de Datos:** Perfecto para bases de datos de relación SQL, caso del proyecto.
- **Adecuado para proyectos académicos:** PHP es adecuado para proyectos web tradicionales, que suelen ser el foco en el Bachillerato Informático principalmente por las razones nombradas, pero con un énfasis en el primer punto.

## **Conclusión**

PHP es una opción sólida para proyectos en desarrollo web por su facilidad, bajo costo, Maleabilidad de desarrollo y rendimiento en tareas comunes. Frente a Node.js y Python, PHP está hablando en un entorno tradicional Web.



Root Group

~ *The Root Group Company* ~

Montevideo, 19 de mayo 2025

## **Selección de SGBD (MySQL/MariaDB)**

### **Introducción:**

Este documento justifica técnica y funcionalmente la elección de MariaDB como sistema gestor de bases de datos para el proyecto Draftosaurus. La selección de un SGBD adecuado es fundamental para garantizar rendimiento, seguridad y compatibilidad con la aplicación web desarrollada en PHP.

### **Análisis comparativo**

Característica	MariaDB	MySQL
Compatibilidad	Compatible con MySQL, fácil migración	Estándar en la industria
Rendimiento	Mejoras en optimización y manejo de hilos	Buen rendimiento, pero menos optimizado
Código abierto	Totalmente open source(GPL)	Licencia dual, con partes cerradas
Seguridad	Cifrado avanzado y actualizaciones rápidas	Seguridad estándar, menos actualizaciones
Innovación	Nuevos motores de almacenamiento y funciones	Menos frecuente en nuevas funciones
Integración PHP	Compatible con conectores MySQL	Nativo para PHP



Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

## **Justificación de la elección de MySQL**

- **Fiabilidad y estabilidad:** MySQL es un SGBD ampliamente probado en la industria, ofreciendo un rendimiento fiable y constante para el proyecto.
- **Soporte y documentación:** Cuenta con una vasta comunidad de usuarios y una extensa documentación oficial y de terceros, lo que facilita la resolución de problemas y el aprendizaje en un entorno académico.
- **Formato nativo JSON:** MySQL soporta el tipo de datos nativo JSON\_TABLE, lo que puede ser útil para manejar estructuras de datos flexibles en el futuro.
- **Familiaridad y uso extendido:** MySQL es uno de los SGBD más utilizados en el mundo, lo que significa que el conocimiento adquirido será altamente transferible a futuros proyectos y entornos laborales, además que en el grupo de proyecto ya se tiene conocimientos previos del mismo.
- **Integración con PHP:** La compatibilidad de MySQL con PHP es nativa y excelente, asegurando una integración fluida y eficiente para el desarrollo de la aplicación web.

## **Conclusión**

MySQL es una elección sólida y pragmática para el proyecto Draftosaurus debido a su probada fiabilidad, amplio soporte y familiaridad en el ámbito del desarrollo Web. Si bien MariaDB ofrece ciertas ventajas en innovación y rendimiento bajo cargas extremas, MySQL es más que suficiente para las necesidades del proyecto Draftosaurus, proporcionando una base estable.

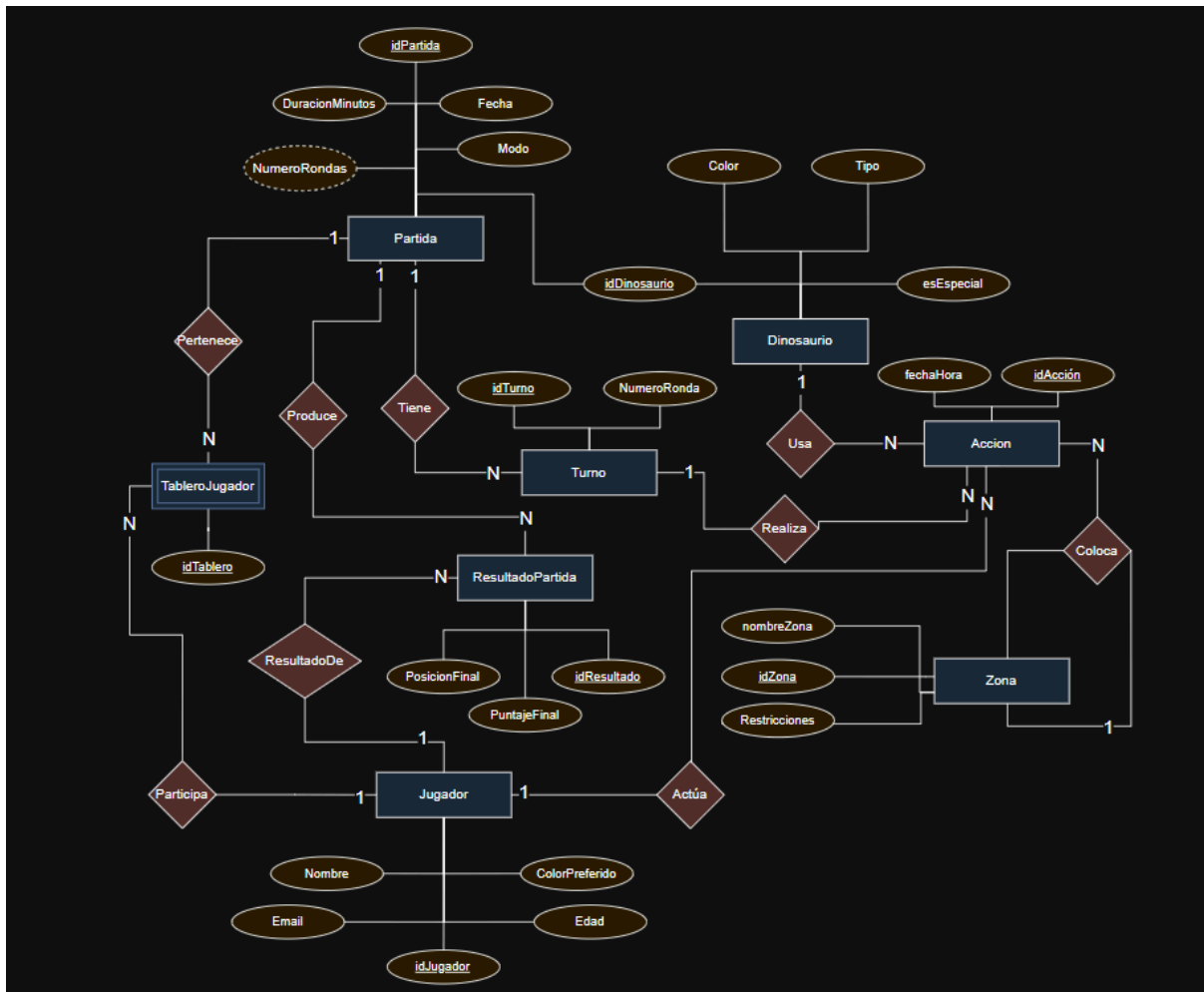


Root Group

~ The Root Group Company ~

Montevideo, 19 de mayo 2025

## DIAGRAMA-ENTIDAD-RELACIÓN (D.E.R)







Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

## **RESTRICCIONES NO ESTRUCTURALES (R.N.E)**

Este documento recopila las Restricciones No Estructurales (R.N.E) del sistema de seguimiento digital del juego Draftosaurus. Las RNE no pueden representarse directamente en el modelo entidad-relación mediante entidades, atributos, relaciones ni cardinalidades, y deben implementarse mediante lógica de aplicación, validaciones, triggers o reglas de negocio.

### **Listado de Restricciones No Estructurales**

#### **RNE (1):**

- ❖ Un jugador solo puede realizar una acción por turno

#### **Tipo:**

- ❖ Restricción de unicidad contextual

#### **Descripción:**

- ❖ Un mismo jugador no puede ejecutar más de una acción (colocar dinosaurio) durante un mismo turno.

#### **Ejemplo:**

- ❖ El jugador Pedro no puede colocar dos dinosaurios en el turno 3.

**Justificación:** El juego establece que cada jugador actúa una sola vez por ronda.

#### **RNE (2):**

- ❖ Un jugador solo puede colocar dinosaurios en su propio tablero

#### **Tipo:**

- ❖ Restricción de integridad semántica

#### **Descripción:**

- ❖ Cada zona en la que un jugador coloca un dinosaurio debe pertenecer a su tablero.

#### **Ejemplo:**

- ❖ El jugador Ana no puede colocar un dinosaurio en una zona del tablero de Juan.



Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

**Justificación:**

- ❖ Cada jugador gestiona únicamente su propio tablero.

**RNE (3):**

- ❖ Un mismo dinosaurio no puede colocarse más de una vez en un tablero

**Tipo:**

- ❖ Restricción de unicidad contextual

**Descripción:**

- ❖ Un dinosaurio físico (pieza) no puede ser registrado dos veces por el mismo jugador en la misma partida.

**Ejemplo:**

- ❖ El jugador Diego no puede colocar el mismo dinosaurio verde dos veces en su tablero.

**Justificación:**

- ❖ Evita duplicaciones que violen las reglas del juego físico.

**RNE (4):**

- ❖ Solo puede haber un jugador activo por turno

**Tipo:**

- ❖ Restricción de exclusividad

**Descripción:**

- ❖ Cada turno tiene exactamente un jugador activo.

**Ejemplo:**

- ❖ En el turno 5, no puede haber dos jugadores actuando a la vez.

**Justificación:**

- ❖ Refleja el turno secuencial del juego.

**Formalización:**

- ❖ Atributo único: idJugadorActivo en Turno
- ❖ Validación: idJugadorActivo debe pertenecer a los jugadores de la partida



Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

**RNE (5):**

- ❖ El número de rondas depende del modo de juego

**Tipo:**

- ❖ Restricción de dependencia condicional

**Descripción:**

- ❖ La cantidad de rondas de una partida se define según el modo configurado.

**Ejemplo:**

- ❖ En modo "clásico" se deben jugar 12 rondas.

**Justificación:**

- ❖ Mantiene coherencia con las reglas oficiales.

**RNE (6):**

- ❖ El puntaje final debe coincidir con el estado del tablero

**Tipo:**

- ❖ Restricción de consistencia calculada

**Descripción:**

- ❖ El puntaje final registrado debe ser el resultado de aplicar las reglas de puntuación al contenido del tablero.

**Ejemplo:**

- ❖ Si un jugador tiene 3 dinosaurios del mismo tipo en una zona con bonificación, el sistema debe reflejar correctamente ese puntaje.

**Justificación:**

- ❖ Garantiza resultados legítimos.

**RNE (7):**

- ❖ Cada jugador tiene un solo tablero por partida

**Tipo:**

- ❖ Restricción de unicidad

**Descripción:**



Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

- ❖ Un jugador no puede tener más de un tablero en una misma partida.

#### **RNE (8):**

- ❖ Las acciones deben registrarse en orden cronológico

#### **Tipo:**

- ❖ Restricción temporal

#### **Descripción:**

- ❖ Dentro de cada turno, las acciones deben registrarse con timestamps consistentes.

#### **Ejemplo:**

- ❖ La acción con fechaHora posterior no puede estar antes que otra dentro del mismo turno.

#### **Justificación:**

- ❖ Mantiene el orden real de los eventos.

#### **Formalización:**

- ❖ Ordenar Accion.fechaHora ASC dentro de cada Turno

#### **RNE (9):**

- ❖ Solo se permite un resultado por jugador y partida

#### **Tipo:**

- ❖ Restricción de unicidad

#### **Descripción:**

- ❖ Cada jugador debe tener un único resultado registrado por partida.

#### **RNE (10):**

- ❖ No se puede registrar un resultado si el jugador no participó

#### **Tipo:**

- ❖ Restricción existencial

#### **Descripción:**

- ❖ Un jugador no puede tener resultado si no participó en al menos una acción de la partida.

#### **Justificación:**



Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

- ❖ No se puede puntuar a un jugador inactivo.

#### **RNE (11):**

Número de jugadores debe estar entre 2 y 5

##### **Tipo:**

- ❖ Restricción de límite cardinal

##### **Descripción:**

- ❖ Cada partida debe tener entre 2 y 5 jugadores.

##### **Justificación:**

- ❖ Son las reglas estándar del juego.

#### **RNE (12):**

- ❖ Cada acción debe incluir obligatoriamente un dinosaurio y una zona

##### **Tipo:**

- ❖ Restricción de completitud

##### **Descripción:**

- ❖ No se puede registrar una acción sin información completa.

#### **RNE (13):**

- ❖ Restricciones especiales por tipo de zona

##### **Tipo:**

- ❖ Regla de negocio contextual

##### **Descripción:**

- ❖ Algunas zonas imponen restricciones particulares (por ejemplo, solo un dinosaurio por color).



Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

## **Pasaje a Tabla**

Jugador(idJugador, nombre, edad, email, colorPreferido)

PK → idJugador

Partida(idPartida, fecha, duracionMinutos, modo, numeroRondas)

PK → idPartida

Dinosaurio(idDinosaurio, tipo, color, esEspecial)

PK → idDinosaurio

Zona(idZona, nombreZona, restricciones)

PK → idZona

TableroJugador(idTablero, idJugador, idPartida)

PK → idTablero

FK → idJugador referencia a Jugador(idJugador)

FK → idPartida referencia a Partida(idPartida)

Turno(idTurno, idPartida, numeroRonda, idJugadorActivo)

PK → idTurno

FK → idPartida referencia a Partida(idPartida)

FK → idJugadorActivo referencia a Jugador(idJugador)

Accion(idAccion, idTurno, idJugador, idDinosaurio, idZona, fechaHora)

PK → idAccion

FK → idTurno referencia a Turno(idTurno)

FK → idJugador referencia a Jugador(idJugador)

FK → idDinosaurio referencia a Dinosaurio(idDinosaurio)

FK → idZona referencia a Zona(idZona)



Root Group

~ *The Root Group Company* ~  
*Montevideo, 19 de mayo 2025*

ResultadoPartida(idResultado, idPartida, idJugador, puntajeFinal, posicionFinal)

PK → idResultado

FK → idPartida referencia a Partida(idPartida)

FK → idJugador referencia a Jugador(idJugador)

- **Todas las tablas están en 3FN**

- No hay atributos multivaluados ni compuestos.
- No hay dependencias parciales (todo atributo depende de la clave completa).
- No hay dependencias transitivas (atributos dependen directamente de la clave primaria).



Root Group

~ *The Root Group Company* ~

Montevideo, 19 de mayo 2025

## Formato de convención de Commits

**Título:** si es un avance (update), si es nuevo (create, initial), finalización en el caso de ser necesitado (finishing, done with ...) en el caso de cambios (refactor) - seguido de la parte que se modificó, agregó o creó.

**Descripción:** punteada con mínimas explicaciones concretas.

**Extras:** en caso de ser varios cambios en un mismo commit agregar un (feat) seguido del nombre del archivo para saber a que se hace referencia





Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

## **Especificación de versiones**

Software	Nombre	Versión	Expiración	Justificación
Máquina Virtual	Oracle VirtualBox	V 7.1.8	N/A	Se necesitaba una base para poder colocar la ISO
Emulador Local	XAMPP	V 8.0.30	N/A	Indispensable para poder usar el localhost para pruebas de la web en local
FrameWork	BootStrap	V 5.3.0	N/A	Es una tecnología que es muy efectiva para el Front-End, permitiendo menor tiempo de desarrollo
Lenguaje de Programación	PHP	V 8.4 31	diciembre 2026	Es la última versión de PHP, permite hacer funcionalidades de la mejor manera posible, ante las tecnologías usadas en el proyecto
Bases de datos	MySQL	V 8.4.4	abril 2032	Es un lenguaje de base de datos basado en relacionamiento de datos, ideal para tipos de proyectos como el nuestro, en donde se necesitan relacionar datos de manera SQL
Diseño Gráfico	Figma	V 116.14.4	N/A	Líder en diseños, sumamente usado en el desarrollo Front-end empresarial y perfectamente aplicable



Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

Servidor	Fedora server	V 42	abril 2026	Un requerimiento no funcional que sea Red Hat, se eligió esa distribución por experiencia con la misma
Sistema de control de versiones	Git	V 2.45.1	31 diciembre 2026	Es el sistema de control de versiones más utilizado en entornos de desarrollo, permite subir archivos y volver a anteriores versiones
Lenguaje de Programación	Javascript	V ECMAScript 2023	N/A	Es un lenguaje de programación que se utiliza para crear contenido web dinámico e interactivo.
Lenguaje de hipertexto de texto	HTML	HTML5	N/A	HTML5 es la versión más reciente y ampliamente actualizada, utilizada para estructurar y presentar contenido en la web.
Hojas de Estilo	CSS	CSS3	N/A	CSS3 es la última evolución de las hojas de estilo
Entorno de Desarrollo Integrado (IDE)	Trae IDE	V 2.45.1	31 diciembre 2026	Es un entorno de desarrollo similar a Visual Studio de control de versiones Git



Root Group

~ *The Root Group Company* ~  
*Montevideo, 19 de mayo 2025*

## **Fundamento de Git y GitHub**

Usar Git y GitHub en cualquier proyecto de ámbito tecnológico es fundamental por varias razones.

Primero, Git nos permite tener un control total sobre las versiones del código. Esto significa que si cometemos un error o queremos volver a una versión anterior, lo podemos hacer sin



Root Group

~ *The Root Group Company* ~

*Montevideo, 19 de mayo 2025*

problemas. Además, cada cambio queda registrado, así que siempre sabemos quién hizo qué y cuándo.

Por otro lado, GitHub es una plataforma que nos facilita mucho el trabajo en equipo. Podemos subir nuestro proyecto a la nube y todos los integrantes del grupo pueden acceder al código desde cualquier lugar. También es muy útil para organizar el trabajo, porque permite crear ramas para nuevas funciones o arreglos, y después unirlos al proyecto principal cuando estén listas. Así, evitamos pisarnos el trabajo entre compañeros.

Otra cosa importante es que GitHub nos ayuda a documentar el proyecto, por ejemplo usando archivos README, que resumen de forma concisa de qué trata el proyecto y cómo usarlo. En resumen, usar Git y GitHub nos da organización, seguridad y una forma profesional de trabajar en proyectos tecnológicos, algo que es cada vez más importante en el mundo laboral.

#### Fundamentación de Tecnologías Aplicadas a Draftosaurus

A continuación la fundamentación de por qué las tecnologías elegidas son ideales para Draftosaurus.

❖ **PHP:** Es como el motor que hace funcionar la parte de atrás de nuestra aplicación web. Nos va a servir para manejar la lógica del servidor, procesar la información que viene de la base de datos y enviar los datos a la parte que ve el usuario.

❖ **HTML:** Es el esqueleto de nuestra página web. Con HTML vamos a armar la estructura básica: dónde va el título, los botones, las imágenes de los dinosaurios, y todo lo que el



Root Group

~ *The Root Group Company* ~

*Montevideo, 19 de mayo 2025*

usuario va a ver en su navegador. Sin HTML, no hay página.

❖ **JavaScript:** Este es el que le da vida al esqueleto. Con JavaScript, la página se vuelve interactiva. Por ejemplo, JavaScript va a manejar que los jugadores puedan arrastrar y soltar dinosaurios, que se muestren mensajes, que haya animaciones o que la interfaz reaccione a lo que hace el usuario. Hace que la página no sea solo texto e imágenes, sino que "se mueva".

❖ **Figma:** Antes de empezar a programar, necesitamos un dibujo de cómo va a ser nuestra aplicación o juego. Figma es como nuestro cuaderno de bocetos digital. Aquí diseñamos la interfaz de usuario: cómo se ve cada pantalla, dónde van los botones, los colores, las fuentes. Es clave para tener una idea clara antes de escribir una sola línea de código y que todos en el equipo estén de acuerdo con el diseño.

❖ **Bootstrap:** Una vez que tenemos el diseño en Figma, Bootstrap nos ayuda a que la parte visual de la web se vea bien y sea "responsive". Es decir, que se vea bien tanto en una computadora grande como en un celular. Bootstrap nos da un montón de estilos predefinidos (botones, menús, rejillas) que hacen que nuestra aplicación se vea moderna y profesional sin tener que escribir todo el código CSS desde cero.

❖ **XAMPP:** Este es un paquete todo en uno que nos permite tener un servidor web en nuestra propia computadora, sin necesidad de estar conectados a internet. Con XAMPP, podemos probar nuestra aplicación web (PHP, HTML, JavaScript) y nuestra base de datos MySQL como si ya estuvieran publicadas, pero sin que nadie más las vea. Es perfecto para desarrollar y hacer pruebas tranquilamente.

❖ **Git:** Es nuestro sistema de control de versiones. Con Git, cada vez que hacemos un cambio en el código (ya sea en PHP, HTML o JavaScript), queda un registro. Si algo sale mal, podemos volver a una versión anterior. Es como tener un "guardado" constante de todo el proyecto. Esto es fundamental cuando varias personas trabajan en lo mismo para evitar pisarse los cambios.



Root Group

~ *The Root Group Company* ~

*Montevideo, 19 de mayo 2025*

❖ **MySQL:** Esta es nuestra base de datos. Draftosaurus necesita guardar información de jugadores, las puntuaciones, los tipos de dinosaurios, entre otros datos, MySQL es el lugar donde se va a guardar todo de forma organizada. PHP se va a conectar con MySQL para guardar y leer la información que necesita.

❖ **Fedora Server:** Para que nuestro juego esté disponible en internet para todo el mundo, necesitamos un servidor. Fedora Server es un sistema operativo Linux diseñado para eso. Aquí es donde vamos a instalar PHP, MySQL y donde vamos a subir todos nuestros archivos de Draftosaurus para que la gente pueda acceder a ellos desde cualquier lugar.

## **Fundamentación de Visual Studio Code**

Visual Studio Code (VS Code) es una herramienta que ofrece muchas ventajas para hacer proyectos tecnológicos y hace que el trabajo diario sea mucho más sencillo. A simple vista parece un editor básico, pero lo que realmente lo hace especial son las extensiones y complementos que se pueden agregar para adaptarlo a lo que necesitamos.



Root Group

~ **The Root Group Company** ~

*Montevideo, 19 de mayo 2025*

Una de las cosas más valiosas de VS Code es que es ligero y rápido, por lo que no hace falta tener una computadora muy potente para usarlo bien, como es el caso de Xcode. Esto es importante porque no todos tenemos equipos nuevos o muy caros. Además, es gratis y funciona en Windows, Linux y Mac, así que todos los compañeros del grupo pueden usarlo sin problemas.

VS Code acepta muchos lenguajes de programación, desde los más comunes como Python, JavaScript o C++, hasta otros menos usados. Esto nos da la libertad de trabajar en diferentes proyectos sin tener que cambiar de programa. También tiene muchas extensiones que podemos instalar para mejorar la experiencia: por ejemplo, para corregir errores, para usar Git o para que el código se vea mejor.

Hablando de Git, VS Code se lleva muy bien con esta herramienta. Podemos hacer cambios, crear ramas y arreglar problemas directamente desde el editor, sin tener que usar la consola o programas aparte. Esto hace que el trabajo sea más simple y que el proyecto quede más ordenado.

La interfaz de VS Code es clara y fácil de entender, con menús y paneles que se acomodan a lo que estamos haciendo. Por eso es fácil de usar, incluso para los que recién empiezan. Además, tiene una función para encontrar y corregir errores dentro del mismo programa, lo que ahorra mucho tiempo y evita frustraciones.

Por último, VS Code tiene una comunidad muy activa. Esto quiere decir que siempre hay nuevas extensiones, tutoriales y ayuda disponible, lo que nos ayuda a seguir aprendiendo y mejorando.



Root Group

~ *The Root Group Company* ~  
*Montevideo, 19 de mayo 2025*

## **Documentación Técnica del Frontend**

**Proyecto:** Sistema Informático de Gestión de Partidas para Draftosaurus (S.I.G.P.D.)

Componente: Interfaz de Usuario (Frontend)

### **1. Introducción**

El frontend del sistema fue desarrollado en HTML5, CSS3 y JavaScript, con integración de Bootstrap 5.3 para diseño responsivo. Se priorizó una interfaz clara, amigable y funcional, emulando el estilo visual del juego físico Draftosaurus, pero adaptado a un entorno web.





Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

## 2. Justificación de Tecnologías Seleccionadas

### HTML5

- Estructura semántica para accesibilidad y mantenibilidad.
- Uso de etiquetas como <section>, <header>, <main>, <nav>, <footer>.

### CSS3 + Bootstrap 5.3

- Bootstrap 5.3 permitió crear una experiencia responsiva y mobile-first.
- Utilización de clases predefinidas para diseño rápido y ordenado.
- Estilos personalizados para replicar la estética del juego y apps móviles.

### JavaScript

- Controla la lógica de interacción, renderizado dinámico y reglas del juego.
- Modular y escalable.
- Usa eventos, animaciones y validaciones.
- Implementación de drag-and-drop y control de turnos.

### Imágenes como componentes interactivos

- Cada recinto es una imagen recortada programada con lógica propia.
- Aplicación de restricciones según dado y especie.
- El río tiene lógica diferenciada para colocación forzada.

## 3. Estructura General del Frontend

### inicio.html

- Página de bienvenida con navegación clara y estilo visual moderno.

### partida.html

- Tablero con zonas recortadas.
- Dado funcional e interactivo.
- Drag-and-drop de dinosaurios.
- Validación de reglas por recinto y dado.
- Lógica de turnos.



Root Group

~ *The Root Group Company* ~  
Montevideo, 19 de mayo 2025

## **facebook.html**

- Pantalla final de simulación con estética de redes sociales.
- Tabla de resultados o botones de compartir.

## **4. Diseño Responsivo**

- Bootstrap + media queries.
- Botones accesibles.
- Tablero adaptable a dispositivos móviles y tablets.

## **5. Experiencia de Usuario (UX)**

- Control manual de acciones por el jugador.
- Estilo visual claro, colorido y sin sobrecargas.
- Elección libre de dinosaurios y animaciones ligeras.

## **6. Modularidad y Escalabilidad**

- Preparado para añadir recintos, modo multijugador y conexión a backend.
- Separación clara entre vista, lógica y datos.

## **7. Futuras mejoras previstas**

- Transiciones de turnos.
- Configuración previa de partida.
- Puntuación en tiempo real.
- Conexión con base de datos para estadísticas.

## **8. Conclusión**

Este frontend emula fielmente la experiencia de Draftosaurus. Las tecnologías fueron seleccionadas por su compatibilidad, eficiencia y facilidad de desarrollo, cumpliendo con los requerimientos técnicos del proyecto.



Root Group

~ ***The Root Group Company*** ~  
*Montevideo, 19 de mayo 2025*

## **Hoja Testigo**