

# Justificación Tecnológica

## TECNOLOGÍAS:

En la realización de nuestra aplicación web, basada en el juego de mesa “*Draftosaurus*”, utilizaremos las tecnologías HTML, CSS, JavaScript y frameworks como Bootstrap por el lado del frontend. Para el backend tendremos un servidor en el que correrá Red Hat, un sistema operativo linux, guardando entre otras cosas, nuestra base de datos MySQL. Se implementará PHP como lenguaje para manejar la lógica de este servidor. A continuación una breve descripción que justifica y evidencia el empleo de este stack tecnológico:

### **HTML HyperText Markup Language (Lenguaje de Marcado de Hipertexto)**

Es el estándar fundamental para la creación de páginas, sitios y aplicaciones web. Es un lenguaje de marcado que estructura y organiza el contenido de la aplicación, como textos, imágenes, formularios, tablas u otros elementos. Su sintaxis fácil de aprender y utilizar agiliza los tiempos de desarrollo. Nuestro objetivo es crear interfaces web accesible desde cualquier navegador y dispositivo, por tanto, HTML es indispensable para crear la estructura básica que permitirá al usuario interactuar con la aplicación.

### **CSS Cascading Style Sheets (Hojas de Estilo en Cascada)**

Se usa para definir la presentación visual de la página web creada con HTML. Permite controlar el diseño, los colores, las fuentes y la disposición de los elementos en la interfaz de usuario. Facilita la creación de una interfaz limpia, ordenada y atractiva que respeta las normas de diseño visual de la aplicación, lo que mejora significativamente la experiencia del usuario. Además, permite separar el contenido HTML de su estilo, logrando un código más organizado y fácil de mantener.

### **JavaScript**

JavaScript es un lenguaje de scripting popular y esencial para el desarrollo web. Permite agregar interacciones y funcionalidades dinámicas a las páginas. Con él se pueden programar respuestas a las acciones del usuario como click en botones, crear efectos visuales, animaciones y transiciones que mejoran la UI/UX. Maneja validación de entradas. Es ampliamente soportado

por todos los navegadores web. Es un lenguaje relativamente fácil de aprender. Permite manipular el Modelo de Objetos de Documento (DOM) para actualizar contenido dinámicamente, modificar la estructura y el estilo de la página web en tiempo real. Además JavaScript es un lenguaje utilizado para frontend y backend.

## PHP como lenguaje backend:

PHP es un lenguaje de programación interpretado del lado del servidor, de código abierto y diseñado especialmente para el desarrollo web. Fue creado por Rasmus Lerdorf al principio, y publicado en 1995, para complementar todo lo que se estaba haciendo en el frontend de la web en aquella época. Actualmente corre aproximadamente en el 76% de los servidores cuyo lenguaje del lado backend se conoce. Entre los principales sitios desarrollados con PHP se encuentran WordPress, Wikipedia y en sus inicios, Facebook. Una parte significativa del ecosistema web actual lo emplea como parte de su arquitectura. A diferencia de lenguajes como JavaScript, que se ejecutan principalmente en el navegador, PHP es un lenguaje de scripting que se ejecuta en el servidor. Esto significa que el código fuente no se muestra en un navegador web, sino que se ejecuta a nivel de servidor con un intérprete. De esta manera, el script se procesa en el backend y el resultado se reenvía al dispositivo. Para nuestro proyecto utilizaremos la versión 8.4. Entre las ventajas que tiene PHP como lenguaje backend se encuentra:

1) **Costo:** Al ser de código abierto es gratuito y disponible para todos los usuarios, ahorrando en licencias costosas y ventajoso para startups o pequeños proyectos que buscan maximizar su presupuesto sin sacrificar calidad.

2) **Compatibilidad:** Es un lenguaje multiplataforma, significando compatibilidad con todos los sistemas operativos más importantes como Linux, Windows y MacOS. Fue diseñado específicamente para ejecutarse en entornos web, especialmente con servidores Nginx y Apache, este último es la opción habitual, ofreciendo más posibilidades de configuración además de un alto nivel de portabilidad. Motivo clave por el cual PHP es tan popular sería por formar parte del stack tecnológico LAMP (Linux, Apache, MySQL y PHP), utilizado habitualmente en internet.

3) **Flexibilidad:** Destaca por su posibilidad de integración en frontend con tecnologías como HTML, JavaScript, frameworks modernos y AJAX, creando sitios web dinámicos e interactivos. Gestiona eficientemente bases de datos debido a su soporte integrado y maduro para MySQL, PostgreSQL, SQLite, SQL Server, entre otros. Esto facilita el almacenamiento y recuperación de

información. Es altamente compatible con herramientas como Docker para crear entornos de desarrollo consistentes. Gestiona de manera óptima el sistema de archivos, cualidad importante para un lenguaje backend. Esto permite realizar operaciones de lectura y escritura en cualquier fichero de las carpetas del servidor donde PHP tenga los permisos suficientes, guardando por ejemplo información de usuarios.

4) **Grupos objetivos:** Está dirigido a programadores experimentados y principiantes. Su curva de aprendizaje reducida, uso y sintaxis accesible lo convierten en una opción viable, permitiendo una implementación ágil a la hora de realizar un proyecto. Esto lo hace un lenguaje fácil para desarrollo web.

5) **Rendimiento:** Aunque la velocidad puede variar según la implementación, las versiones, especialmente desde PHP 7, introducen optimizaciones que mejoran la capacidad de procesamiento. Los datos y bases de datos se cargan rápidamente. Además las características orientadas a los objetos se manejan en muy cortos intervalos de tiempo. Esto es fundamental para aplicaciones web que requieren respuestas de ejecución rápidas.

6) **Seguridad:** En el pasado fue criticado por fallos de seguridad, debido a malas prácticas y estrategias poco sólidas para proteger las aplicaciones. Este panorama cambió. Actualmente PHP incluye funciones nativas para gestionar cifrado, hashing a contraseñas, validación y sanitización de datos entrantes. Esto protege de ataques como la Inyección SQL, prevención de XSS, logrando configuraciones seguras en el servidor.

7) **Ecosistema:** El ecosistema de PHP es uno de los más contundentes y extensos en el desarrollo web, abarcando desde potentes frameworks como Laravel, Symfony, CodeIgniter y CakePHP hasta CMS(Sistema de Gestión de Contenidos) como WordPress y Drupal, permitiendo crear blogs o plataformas de comercio a escala empresarial. Cuenta con un sistema de gestión de dependencias moderno y eficiente, Composer es su nombre, que facilita la integración de miles de librerías disponibles de terceros. Dispone de herramientas robustas para testing (PHPUnit), depuración (Xdebug) y desarrollo local (Laravel Valet, Homestead y XAMPP).

8) **Comunidad:** La comunidad activa de desarrolladores de PHP, impulsó enormemente la evolución del lenguaje con gran cantidad de documentación oficial en diferentes idiomas y recursos de aprendizaje. A partir de la versión 3 la comunidad empezó a participar en el proyecto, incorporando funcionalidades integradas que permitían trabajar con una amplia cantidad de bases de datos, corrigiendo errores y desarrollando herramientas. Recientemente

en las versiones (8.x) aportaron al compilador JIT(Just-In-Time Compiler), optimizando aún más el rendimiento para tareas de cálculo intensivo y procesamiento en tiempo real. Además, impulsaron la integración progresiva del tipado fuerte, permitiendo mayor robustez y detección temprana de errores durante el desarrollo. expandiendo significativamente este popular lenguaje.

## **Ventajas de PHP ante alternativas:**

Al igual que PHP, existen otros lenguajes de programación que se usan por el lado del servidor.

### **Ventajas ante Java:**

Un claro ejemplo es Java como lenguaje backend, utilizado para construir aplicaciones de escritorio y web. Java es más rápido y funcional. No obstante, es difícil de utilizar, lo que puede suponer un mayor consumo de tiempo y recursos. PHP por su lado ofrece ciclos de implementación más rápidos y fácil de mantener debido a su curva de aprendizaje. En entornos empresariales con recursos limitados, una clara ventaja es usar PHP para aplicaciones web. Es una opción más económica y tiene gran disponibilidad de recursos.

### **Ventajas ante Python:**

Python es el lenguaje de programación más popular del mundo actualmente. Es utilizado en diversas áreas como ciencia de datos, aprendizaje automático y desarrollo web. Ambos cuentan con una gran comunidad, son de código abierto y abundante documentación. PHP tiene algunas ventajas claras. Este lenguaje no tiene mucha dificultad a la hora de instalarse, al contrario de Python, si bien en linux se puede instalar de forma rápida, no sucede lo mismo en Windows o MacOS. En cuanto a rendimiento y velocidad, es superior a Python, sobre todo en las últimas versiones puede ser hasta tres veces más rápido. Además PHP tiene mejor integración con servidores web y mayor disponibilidad en hostings compartidos.

## **Sistema gestor de Base de Datos (DBMS)**

Las bases de datos son los repositorios de información esencial para todas las aplicaciones de software. Para nuestra aplicación utilizaremos MySQL como sistema gestor de base de datos relacionales. Este software es de código abierto el cual utiliza SQL para almacenar y gestionar datos. Su fiabilidad, rendimiento optimizado, escalabilidad, seguridad y facilidad de uso son motivos para implementarlo en nuestro proyecto. Como base de datos relacional, MySQL almacena datos en tablas de filas y columnas organizadas en esquemas. Un esquema define cómo se organizan y almacenan los datos y describe la relación entre varias tablas. Con este

formato, los desarrolladores pueden almacenar, recuperar y analizar fácilmente muchos tipos de datos, incluidos texto simple, números, fechas, horas, entre otros. Otros puntos fuertes de esta base de datos son:

### **Compatibilidad directa con PHP y stack LAMP**

Este gestor se integra directamente con PHP, debido a su compatibilidad nativa con extensiones como mysqli y PDO\_MySQL, reduciendo la complejidad del desarrollo backend. Además su adopción en el stack LAMP lo convierte en una opción estándar y utilizada en entornos web. PHP y este stack evidencian la elección de MySQL para nuestro S.I.G.P.D.

### **Facilidad de uso y administración**

Cuenta con herramientas gráficas (como phpMyAdmin o MySQL Workbench) y utilidades de línea de comandos potentes que simplifican tareas de backup, replicación, monitoreo y ajustes de rendimiento. Esto reduce el tiempo de puesta en marcha y mantenimiento.

### **Seguridad robusta**

Incluye soporte para autenticación fuerte, cifrado de conexiones y contraseñas, gestión de permisos y roles de usuario, permitiendo construir bases de datos seguras en ambientes productivos.

### **Comunidad**

Al ser un sistema open-source con respaldo de Oracle, MySQL cuenta con un amplio soporte comunitario muy activo, abundante documentación oficial y de terceros, y soporte profesional si se requiere en entornos críticos.

## **Framework frontend: [Bootstrap](#)**

*Bootstrap* es una excelente opción para el desarrollo frontend, ya que proporciona un conjunto completo y listo para usar de componentes responsivos, estilos consistentes y utilidades CSS, lo que permite construir interfaces modernas, limpias y adaptables a distintos tamaños de pantalla sin necesidad de diseñar desde cero. Entre los beneficios de usar este framework se encuentran:

### **Aceleración del desarrollo**

Bootstrap proporciona componentes predefinidos y estilos CSS, lo que permite a los desarrolladores crear sitios web más rápidamente sin tener que escribir mucho código. Incluye elementos como botones, formularios, alertas, tarjetas, carruseles, etc., que pueden integrarse fácilmente en cualquier sitio web. Además, brinda estilos y funcionalidad adicional para personalizar los componentes y el diseño general.

## Diseños responsivos

Su enfoque "mobile-first" garantiza que los sitios web se adapten automáticamente a diferentes tamaños de pantalla, mejorando la experiencia del usuario en dispositivos móviles.

## Facilidad de aprendizaje

Bootstrap tiene una curva de aprendizaje relativamente fácil, y su amplia documentación y comunidad de apoyo hacen que sea fácil aprender y usar.

## Sistema GRID

El sistema grid de Bootstrap es una estructura de diseño que contiene 12 columnas las cuales permiten organizar y distribuir el contenido de manera flexible y responsive. Utiliza una combinación de filas (.row) y columnas (.col), que se adaptan automáticamente al tamaño de la pantalla gracias a clases predefinidas como .col-sm-, .col-md-, .col-lg- y .col-xl-. Estas clases indican cuántas columnas debe ocupar un elemento en distintos puntos de ruptura (breakpoints), permitiendo crear interfaces que se ajustan fluidamente desde dispositivos móviles hasta pantallas grandes. Además, el grid admite alineación, espaciado (g- para gutters) y orden en el contenido, facilitando la creación de layouts complejos sin escribir CSS adicional.

## Herramienta para control de versiones y colaboración

Un sistema de control de versiones (VCS) registra los cambios realizados en un archivo o conjunto de archivos para poderlos recuperar posteriormente permitiendo regresar a versiones anteriores de los ficheros e incluso proyectos completos. Además, se utiliza para comparar cambios a lo largo del tiempo y ver quien modificó algo que puede estar dando problemas. Usar un VCS también significa que, si se rompe el código o se pierden archivos, será posible recuperarlos fácilmente.

# Git

Entre los VCS que existen, seleccionamos *Git* como sistema de control de versiones. Tiene la particularidad de ser distribuido. Esto significa que, al descargar la última copia de los archivos, se replica completamente el repositorio de manera local, almacenando un historial completo del proyecto. Si un servidor deja de funcionar, cualquiera de los repositorios disponibles en los clientes puede ser copiado al servidor para restaurarlo. Cada clon es una copia exacta de todos los datos. Este software desarrollado por Linus Torvalds, facilita el trabajo colaborativo, permite trabajar sin conexión y realizar operaciones como commits, ramas, fusiones (*merges*) y reversiones de forma eficiente. Este sistema está pensado para manejar grandes volúmenes de archivos eficientemente de manera rápida debido a su arquitectura basada en instantáneas. Git permite crear múltiples ramas independientes para probar nuevas funcionalidades sin afectar el código principal, y luego integrarlas de forma segura. Lo utilizaremos ampliamente en las entregas y avances de nuestro proyecto, facilitando el trabajo en equipo y la creación del código en nuestra aplicación web.

# Github

GitHub es una plataforma en línea que permite almacenar, compartir y colaborar en proyectos que usan Git como sistema de control de versiones. Actúa como un repositorio remoto donde los desarrolladores pueden subir su código, trabajar en equipo, llevar historial de cambios y contribuir a proyectos de forma organizada. Entre sus puntos fuertes destaca:

## Control de versiones centralizado

Actúa como un repositorio remoto donde se centraliza el código y el historial de cambios, permitiendo a todos los colaboradores sincronizar su trabajo. Aunque Git es distribuido, GitHub proporciona un punto de referencia común, lo cual evita confusiones sobre cuál es la versión más actual del proyecto. Esto mejora la coordinación entre desarrolladores y asegura que todos trabajen sobre la misma base de código, minimizando errores como sobrescritura de archivos o pérdida de cambios.

## Gestión de ramas

Permite la creación, visualización y gestión de ramas de forma clara y eficiente. Cada desarrollador puede trabajar en una rama independiente, ya sea para una nueva funcionalidad o corrección de errores. Esto no afecta el código del entorno principal (main o master). Una vez que los cambios han sido probados, se puede fusionar la rama con el proyecto principal.

mediante un *pull request*, manteniendo un desarrollo limpio, organizado y sin interrupciones. Esto fomenta una arquitectura de trabajo modular y segura.

## **Historial transparente**

Cada *commit* en GitHub registra el contenido modificado, el autor, la hora y un mensaje descriptivo, lo cual brinda una trazabilidad completa del desarrollo del proyecto. Esto es fundamental para depurar errores y entender el flujo de trabajo.

## **Accesibilidad y comunidad**

Al ser una plataforma en la nube, GitHub puede usarse desde cualquier dispositivo con acceso a internet. Esto lo convierte en una herramienta ideal para equipos distribuidos geográficamente o proyectos de código abierto. Además, alberga una de las comunidades de desarrolladores más grandes del mundo, lo que facilita el aprendizaje, siendo posiblemente el repositorio en donde hay alojado mayor cantidad de código.

# **Entornos de Desarrollo**

Un entorno de desarrollo es el conjunto de herramientas, software y configuración que un programador utiliza para crear, probar y mantener aplicaciones o programas. Nuestro entorno de desarrollo en principio será local, haremos todas las pruebas utilizando XAMPP simulando un servidor. Para escribir el código en PHP, JavaScript, HTML, CSS y Bootstrap, utilizaremos el editor de texto Visual Studio Code. Este es el editor de texto más potente actualmente, debido a su cantidad de extensiones útiles para estos lenguajes, ser multiplataforma y gratuito. Se vincula con repositorios de Git para trabajar en ellos. Permite correr y debuggear los códigos de los archivos que creamos. Otras opciones pueden ser Sublime o IDE como PhpStorm. Este último es pago, aunque tiene una prueba gratuita por 30 días. Utilizaremos un servidor para alojar nuestra app. En este correrá el sistema operativo Red Hat Enterprise Linux en su versión 10. Utilizamos este sistema por ser una distro basada en el kernel de Linux. Además, es orientado a servidor y presenta un alto grado de seguridad. Tiene una versión gratuita oficial a través del programa “Red Hat Developer Subscription” en el cual se puede obtener una ISO para instalar en servidor, sistema host o virtual. Otros S.O para servidor pueden ser Ubuntu Server, Windows Server o CentOS. La base de datos recomendada es MySQL por su compatibilidad con el stack LAMP, el cual usaremos. Otras opciones son PostgreSQL, Maria DB o SQLite.