



S.I.G.P.D.

Programación Full Stack

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: Prestes, Matías

**Fecha de
culminación
14/7/2025**

PRIMERA ENTREGA



ÍNDICE

REPOSITORIO DE GitHub:	2
Análisis de PHP como lenguaje backend	2
Criterios de selección del sistema gestor de base de datos (SGBD)	3
Evaluación de frameworks frontend seleccionados (Bootstrap y complementarios)	5
Selección fundamentada de herramientas para control de versiones y colaboración	6
Recomendaciones de entornos de desarrollo con sus ventajas	7
Cómo probar el proyecto	9

REPOSITORIO DE GitHub:

<https://github.com/JurassiCode/PROYECTO>

Análisis de PHP como lenguaje backend

El lenguaje PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es una tecnología ampliamente utilizada en el desarrollo de aplicaciones web dinámicas. Su diseño orientado al entorno web, su larga trayectoria y su ecosistema robusto lo posicionan como una alternativa vigente frente a otros lenguajes backend como Python, JavaScript, Ruby o Java.

Una de las principales ventajas de PHP es su amplia compatibilidad con servicios de hosting, especialmente en entornos compartidos, lo que reduce los costos y simplifica el despliegue de aplicaciones. A diferencia de lenguajes como Node.js o Django, que suelen requerir configuraciones específicas o entornos dedicados, PHP puede ejecutarse en servidores tradicionales sin mayores requerimientos técnicos. Además, su sintaxis simple y curva de aprendizaje accesible lo convierten en una excelente opción para desarrolladores principiantes. Su integración directa con HTML permite desarrollar aplicaciones web dinámicas sin necesidad de arquitecturas complejas, lo cual acelera el prototipado y facilita el mantenimiento. PHP cuenta con un ecosistema maduro, en el que destacan frameworks como Laravel y Symfony, los cuales proporcionan herramientas modernas para enrutamiento, seguridad, ORM (Object-Relational Mapping), validaciones y más. Laravel, en particular, ofrece una sintaxis limpia, documentación extensa y una comunidad activa que contribuye constantemente con recursos y paquetes. Otra ventaja clave es que PHP fue diseñado desde sus inicios como un lenguaje orientado a la web.



Por ello, incorpora de forma nativa funcionalidades como manejo de sesiones, envío de correos electrónicos, procesamiento de formularios y conexión a bases de datos, lo que permite construir aplicaciones completas sin necesidad de librerías externas. En cuanto a la seguridad, si bien PHP fue históricamente criticado por malas prácticas en su implementación, hoy en día los frameworks modernos incorporan medidas de protección contra amenazas comunes como inyecciones SQL, ataques XSS y CSRF, entre otras.

En conclusión, PHP continúa siendo una opción sólida y vigente en el desarrollo backend, especialmente en proyectos orientados al entorno web que requieren rapidez de desarrollo, bajo costo de infraestructura y facilidad de mantenimiento.

Criterios de selección del sistema gestor de base de datos (SGBD)

La elección del sistema gestor de base de datos (SGBD) es una decisión estratégica que impacta en múltiples aspectos del desarrollo y funcionamiento de una aplicación. En este proyecto, se seleccionó el SGBD en función de criterios técnicos objetivos, considerando las necesidades específicas de la arquitectura propuesta.

En primer lugar, se prioriza la compatibilidad con el lenguaje backend. Dado que el desarrollo se realiza en PHP, se seleccionó un SGBD que ofreciera integración directa mediante extensiones como mysqli o PDO. MySQL, ampliamente adoptado en entornos PHP, cumple con este requerimiento y permite una implementación fluida de operaciones de base de datos.

También se consideró la facilidad de uso y curva de aprendizaje. MySQL dispone de interfaces gráficas como phpMyAdmin y MySQL Workbench, que simplifican la administración y manipulación de datos, incluso para usuarios con experiencia limitada en bases de datos.



En cuanto al rendimiento, MySQL ha demostrado ser eficiente en entornos web donde predominan operaciones frecuentes de lectura y escritura. Su motor InnoDB permite garantizar integridad referencial mediante claves foráneas, lo cual es esencial en aplicaciones que manejan relaciones entre tablas, como usuarios, partidas y puntuaciones.

Además, se valoró la portabilidad y escalabilidad. MySQL puede desplegarse tanto en servidores locales como en la nube, y funciona en múltiples plataformas, lo cual favorece el crecimiento del proyecto sin necesidad de reestructuraciones importantes.

El modelo de licenciamiento fue otro aspecto considerado. MySQL es un software de código abierto bajo licencia GPL, lo cual permite su uso sin costos asociados, algo crucial en proyectos académicos o con presupuesto limitado.

Finalmente, la seguridad y el soporte comunitario también fueron elementos determinantes. MySQL permite configurar permisos por usuario, restringir accesos y definir políticas de autenticación, y cuenta con una comunidad global activa, así como abundante documentación.

En función de todos estos criterios, se concluye que MySQL es el sistema gestor de base de datos más adecuado para el presente proyecto, por su compatibilidad, robustez, eficiencia y sostenibilidad.



Evaluación de frameworks frontend seleccionados (Bootstrap y complementarios)

Bootstrap

Para el desarrollo frontend del proyecto se seleccionó Bootstrap como framework principal. Esta decisión se basó en su facilidad de uso, su integración rápida en cualquier entorno web, y la extensa documentación que lo acompaña.

Bootstrap ofrece un sistema de grillas responsive, lo que permite adaptar el diseño a distintos tamaños de pantalla sin necesidad de escribir reglas CSS personalizadas. También proporciona una gran variedad de componentes predefinidos (botones, formularios, tarjetas, barras de navegación, entre otros), que simplifican la construcción de interfaces visuales modernas y coherentes.

Además, incluye clases utilitarias que facilitan la personalización del diseño sin salir del HTML, lo que acelera el desarrollo y reduce la dependencia de archivos CSS personalizados.

A nivel técnico, Bootstrap fue una buena elección porque permite enfocarnos más en la lógica y funcionalidad del sistema, sin dejar de lado una apariencia profesional y ordenada.

Hasta el momento, no se han utilizado frameworks o herramientas complementarias, ya que Bootstrap cubre adecuadamente todas las necesidades visuales e interactivas del proyecto en esta etapa.



Selección fundamentada de herramientas para control de versiones y colaboración

Para gestionar el control de versiones y facilitar el trabajo colaborativo durante el desarrollo del proyecto, seleccionamos las herramientas Git y GitHub. A continuación, fundamentamos esta elección

Git

Git es un sistema de control de versiones distribuido, ampliamente utilizado en el desarrollo de software. Nos permite:

Registrar todos los cambios realizados en el código de forma ordenada y cronológica.

Trabajar en paralelo sin sobrescribir los avances de otros integrantes, mediante ramas (branches).

Revertir fácilmente a versiones anteriores en caso de errores o conflictos.

Trabajar sin conexión, ya que cada desarrollador tiene una copia local completa del repositorio.

La elección de Git responde a su eficiencia, velocidad, amplia documentación y al hecho de que es una herramienta estándar en la industria.

GitHub

GitHub es una plataforma web que complementa el uso de Git, ofreciendo:

Almacenamiento remoto del repositorio, accesible desde cualquier lugar.

Gestión de ramas, revisiones de código (pull requests) y seguimiento de issues o tareas.

Visualización del historial de cambios y colaboración entre los miembros del equipo en tiempo real.



Integraciones con otras herramientas como Trello, Slack o GitHub Actions, lo cual puede facilitar la automatización de tareas.

Optamos por GitHub por ser una de las plataformas más utilizadas en el entorno profesional y educativo, lo que también facilita el aprendizaje de buenas prácticas de desarrollo colaborativo.

Recomendaciones de entornos de desarrollo con sus ventajas

Recomendaciones de entornos de desarrollo y sus ventajas

Para el desarrollo del sistema web, se recomienda utilizar los siguientes entornos y herramientas, que se integran de forma eficiente para facilitar el desarrollo, prueba y mantenimiento del proyecto:

1. Visual Studio Code (VS Code)

Ventajas:

Gratuito, liviano y multiplataforma.

Amplia disponibilidad de extensiones (PHP, HTML, Live Server, Git, etc).

Autocompletado inteligente y resaltado de sintaxis.

Integración con terminal y control de versiones (Git).

Vista previa en vivo y depuración integrada.

2. XAMPP / LAMP (Entorno local de servidor web)

Ventajas:

Permite simular un servidor local con Apache, PHP y MySQL sin necesidad de internet.

Fácil instalación y configuración.

Incluye phpMyAdmin para gestionar bases de datos gráficamente.

Ideal para desarrollar y probar antes de desplegar online.



3. phpMyAdmin

Ventajas:

Interfaz web amigable para gestionar MySQL.

Permite crear, modificar y visualizar bases de datos sin necesidad de comandos SQL complejos.

Exporta e importa estructuras y datos fácilmente (útil para backups y migraciones).

4. Navegadores modernos (Chrome, Firefox, Edge)

Ventajas:

Herramientas de desarrollo integradas (DevTools) para depurar código HTML, CSS y JS.

Simulación de dispositivos móviles para test responsive.

Buen soporte para estándares web.

5. Git + GitHub / GitLab

Ventajas:

Control de versiones: permite ver cambios, volver atrás y colaborar en equipo sin pisarse.

Trabajo en ramas para probar nuevas funcionalidades sin romper la versión estable.

Historial completo de desarrollo y colaboración remota.

6. Lenguajes y tecnologías elegidas (HTML, CSS, JS, PHP y MySQL)

Ventajas:

Son tecnologías ampliamente soportadas y con documentación disponible.

PHP y MySQL son ideales para sistemas CRUD y login como el que usamos.

HTML, CSS y JS permiten personalizar totalmente la experiencia de usuario.

Código 100% controlado por el equipo, sin depender de plataformas externas.



Cómo probar el proyecto

1. Requisitos Previos

- PHP ≥ 7.4 con PDO y ext-password habilitado.
- MySQL (o MariaDB)
- Un servidor web (Apache/Nginx) o un paquete todo-en-uno como XAMPP, MAMP o Laragon

2. Copiar los archivos al servidor

Estructura

draftosaurus/

```
|— index.php
|— login.php
|— registro.php
|— dashboard.php
|— welcome.php
|— logout.php
|— database.sql
|— includes/
|   |— connexion.php
|   |— auth.php
|   |— funciones_usuario.php
|— assets/
|   |— css/
|   |— images/
```



3. Configurar la conexión a la base de datos

En includes/conexion.php ajusta:

```
$host = 'localhost';  
$db   = 'draftosaurus';  
$user = 'TU_USUARIO_DB';  
$pass = 'TU_PASSWORD_DB';  
$charset= 'utf8mb4';
```

Importar el archivo.sql en tu manejador de base de datos.

El admin por defecto es:

Usuario: admin1

Contraseña: admin123

4. Iniciar el servidor y probar.