



S.I.G.P.D.

Programación Full Stack

Pétalos Studio

Rol	Apellido	Nombre	C.I	Email
Coordinador	García	Julieta	5.667.504-3	juligarcia007@gmail.com
Sub-Coordinador	Rama	Xavier	5.670.415-3	xavier2rama@gmail.com
Integrante 1	Irrazabal	Joaquin	5.649.943-3	joaquinirrazabal947@gmail.com

Docente: Borba, Gabriela

**Fecha de
culminación
14/07/2025**

PRIMERA ENTREGA



Justificación Tecnológica	2
1. HTML5, CSS3 y JavaScript	2
2. PHP	3
3. MySQL	4
4. Linux	4
5. VirtualBox	4
6. Visual Studio Code	5
7. Bootstrap	5
8. ClickUp	6
9. Lucidchart	6
10. Lovable	7
11. GanttProject	7
Configuración del Entorno de Desarrollo	7
Introducción	7
1. Requisitos de Software	8
2. Instalación y Configuración	8
2.1. Instalación de Linux Mint	8
2.2. Instalación de Apache, PHP y MySQL	9
2.3. Instalación y configuración de Visual Studio Code	9
2.4. Organización del proyecto	10
2.5 Uso diario del entorno en VS Code	10
3. Instrucciones para replicar el entorno	11
Commits	11
4. Modelado de Datos	12
4.1 Modelo Entidad-Relación (DER)	12
4.2 Esquema Relacional Normalizado	13
Entidades:	13
Identificación de Claves Foráneas (FKs)	14
Modelo Entidad Relación (RNE):	15
Enums:	16
Repositorio en GitHub	16



Justificación Tecnológica

1. HTML5, CSS3 y JavaScript

Para el desarrollo de la interfaz web del sistema se utilizó un conjunto de tecnologías fundamentales: HTML5, CSS3 y JavaScript. Estas herramientas son la base del desarrollo web moderno y fueron seleccionadas por su compatibilidad, facilidad de uso y porque forman parte de los contenidos abordados durante el transcurso de todo el bachillerato en la institución.

HTML5 permite definir la estructura y el contenido de las páginas, utilizando etiquetas que facilitan tanto la lectura del código como el posicionamiento en



diferentes dispositivos. CSS3 se utilizará para dar estilo al contenido, permitiendo personalizar la apariencia visual, adaptando el diseño a distintos tamaños de pantalla y mejorando la experiencia del usuario. JavaScript se usará para implementar la interactividad del sistema, permitiendo funcionalidades como selección de dinosaurios, validaciones en el cliente y actualizaciones dinámicas sin necesidad de recargar la página. Su implementación es esencial para simular aspectos del juego original y dar vida al tablero virtual.

2. PHP

PHP fue seleccionado como lenguaje de backend por su simplicidad, amplia adopción en entornos web y facilidad de integración con tecnologías frontend y bases de datos relacionales como MySQL. En el contexto del proyecto Draftosaurus, donde se requiere una lógica de juego basada en reglas simples, validación de turnos, y gestión de partidas en tiempo real básico, PHP resulta ideal por su modelo de ejecución secuencial y su bajo costo de implementación.

Ventajas frente a otras alternativas:

- **Java:** Requiere mayor estructura y tiempo de configuración.
- **Python:** Más versátil, pero frameworks como Django implican una curva de aprendizaje adicional y estructura más rígida.
- **Node.js:** Potente para sistemas en tiempo real, pero con mayor complejidad técnica, innecesaria para el nivel de interacción del sistema.

Las características de PHP permiten centrarse en la lógica del juego, como el control de rondas, distribución de dinosaurios, validación de reglas y almacenamiento de resultados.



3. MySQL

La base de datos será implementada utilizando MySQL, un sistema gestor de bases de datos relacional conocido por su eficiencia, velocidad y compatibilidad con PHP. Se seleccionó por ser una herramienta libre, bien documentada y ampliamente utilizada en el ámbito del desarrollo web.

MySQL permite modelar la información del juego mediante entidades y relaciones normalizadas. A través de su uso se podrá organizar y gestionar datos tales como usuarios, partidas, puntuaciones y configuraciones, asegurando su integridad y disponibilidad. Su sintaxis accesible y su facilidad de instalación dentro del entorno LAMP harán posible una integración sencilla y funcional con el sistema.

4. Linux

Para el entorno del servidor se utilizó el sistema operativo Linux, más precisamente dentro de una distribución montada en una máquina virtual. Linux fue elegido por tratarse de un sistema operativo gratuito, seguro, estable y ampliamente usado en entornos de producción.

El uso de Linux brindará la posibilidad de simular un entorno de servidor real, donde se podrán realizar tareas como la instalación del servidor web Apache, la configuración de puertos, la administración de paquetes y el despliegue del sistema. Su uso también aportará al aprendizaje técnico, permitiendo conocer comandos y herramientas propias de entornos profesionales.

5. VirtualBox

VirtualBox fue la herramienta de virtualización utilizada para montar el servidor. Permitió crear una máquina virtual donde se instaló Linux y se configuró



todo el entorno necesario para ejecutar la aplicación de forma local, como si estuviera en un servidor remoto.

Esta elección se debió a que permite experimentar con instalaciones reales sin necesidad de modificar el sistema operativo principal del equipo. Además, facilitará la prueba del sistema, el despliegue en un entorno controlado y el trabajo en tareas como configuración de red, usuarios y servicios.

6. Visual Studio Code

Visual Studio Code será el editor de código utilizado a lo largo del proyecto. Se eligió por su facilidad de uso, sus múltiples extensiones y soporte para los lenguajes utilizados en el sistema.

Su interfaz amigable, el resaltado de sintaxis, la posibilidad de trabajar con control de versiones mediante Git y su integración con terminales y servidores locales, lo convierten en una opción versátil y eficiente para este proyecto. Además, su bajo consumo de recursos y su compatibilidad con distintos sistemas operativos facilitaron su uso.

7. Bootstrap

Bootstrap fue elegido como el framework principal para el desarrollo frontend de la interfaz de usuario por su eficiencia, versatilidad y facilidad de implementación. Proporciona un sistema de grillas responsive y componentes visuales reutilizables que aceleran el proceso de maquetado y diseño, asegurando una apariencia coherente y profesional. Su enfoque mobile first es clave para garantizar que la interfaz se adapte correctamente a diferentes tamaños de pantalla, optimizando la experiencia del usuario y cumpliendo con la portabilidad necesaria para una aplicación web accesible desde cualquier navegador y dispositivo.



8. ClickUp

ClickUp fue la herramienta utilizada para la planificación del proyecto. Se eligió por su facilidad para crear diagramas de Gantt, asignar tareas, establecer fechas de entrega y realizar seguimientos de las actividades del grupo.

Mediante ClickUp se pudo mantener un orden en el desarrollo del proyecto, respetar las fechas pactadas y visualizar el progreso de cada fase de trabajo. Esta organización fue importante para el cumplimiento de las entregas parciales y para evitar acumulación de tareas.

9. Lucidchart

Para representar la lógica del sistema, se utilizó Lucidchart como herramienta de diagramación. En particular, se empleó para la construcción del árbol de decisiones que representa el comportamiento esperado del sistema frente a distintas acciones del usuario.

Esta herramienta permite visualizar con claridad las reglas del juego y organizar de forma gráfica las posibles rutas que puede tomar una partida. Su uso facilitó tanto la planificación como la implementación del sistema, ya que actuó como guía para programar la lógica del juego en el código fuente.



10. Lovable

Junto con Bootstrap, para el desarrollo del prototipo de la interfaz se seleccionó la herramienta Lovable, gracias a su enfoque centrado en la experiencia del usuario y su facilidad de uso. Esta plataforma permite crear prototipos interactivos de forma rápida, intuitiva y sin requerir conocimientos técnicos avanzados, lo que nos resulta cómodo y práctico para el contexto de la primera entrega de nuestro proyecto.

11. GanttProject

Para complementar el uso de ClickUp, se utilizó también la herramienta GanttProject, un software de planificación de proyectos de código abierto. Se eligió por ser completamente gratuito, no depender de una conexión permanente a internet, y permitir la gestión y almacenamiento local de los cronogramas sin limitaciones de tiempo.

A diferencia de ClickUp, GanttProject ofrece mayor flexibilidad para trabajar de forma offline y mantener un control total sobre los archivos generados, lo cual resulta conveniente para asegurar la continuidad del trabajo en caso de restricciones técnicas. Esta doble herramienta permitió asegurar tanto la colaboración en línea como la preservación local del avance del proyecto.

Configuración del Entorno de Desarrollo

Introducción

Este documento describe el proceso de instalación y configuración del entorno de desarrollo utilizado para el proyecto Pétalos Studio, implementado en Linux Mint 21.1 Vera dentro de



una máquina virtual VirtualBox. Se detallan los requisitos, pasos de instalación, configuración del IDE y organización del proyecto para facilitar su replicación.

1. Requisitos de Software

Componente	Versión Utilizada	Función principal
Sistema Operativo	Linux Mint 22.1 xia Edition (64 bits)	Plataforma base del entorno
Servidor Web	Apache 2.4.58	Procesamiento y entrega de páginas web
Lenguaje de servidor	PHP 8.3.6	Generación de contenido dinámico
Gestor de base de datos	MySQL 8.0.42	Almacenamiento y consulta de datos
Gestor gráfico BD	phpMyAdmin	Administración visual de bases de datos
IDE	Visual Studio Code (última versión)	Edición y depuración del código fuente
Extensiones VS Code	Live Server, Live Share	Servidor local y colaboración en tiempo real
Control de versiones	Git (última versión estable)	Seguimiento de cambios y colaboración

2. Instalación y Configuración

2.1. Instalación de Linux Mint

- Se usó Linux Mint 22.1 xia Edition (64 bits) instalado en VirtualBox.



- Descargar desde: <https://linuxmint.com/download.php>
- Crear máquina virtual con al menos 2 GB RAM, 32 GB disco y habilitar VirtualBox Guest Additions para integración.

2.2. Instalación de Apache, PHP y MySQL

Ejecutar en terminal:

```
sudo apt update
sudo apt install apache2 php libapache2-mod-php php-mysql
mysql-server
sudo systemctl start apache2
sudo systemctl enable apache2
sudo systemctl start mysql
sudo systemctl enable mysql
```

- Verificar que Apache funciona abriendo <http://localhost> en la VM.
- Verificar versión PHP con `php -v`.
- Opcional: instalar phpMyAdmin para gestión visual con `sudo apt install phpmyadmin` y configurarlo para Apache.

2.3. Instalación y configuración de Visual Studio Code

1. Descargar el paquete `.deb` oficial desde: <https://code.visualstudio.com/>
Entrar a la carpeta donde se haya instalado el deb.

Permisos para instalar:



```
sudo chmod 777 (Nombre del archivo).deb
```

Instalar con:

```
sudo dpkg -i (Nombre del archivo).deb
```

```
sudo apt-get install -f
```

2. Abrir VS Code.

3. Instalar solo las extensiones:

- **Live Server** (para lanzar servidor local con recarga automática).
- **Live Share** (para colaboración en tiempo real).

2.4. Organización del proyecto

La carpeta del proyecto está ubicada en el directorio raíz del servidor web Apache:

```
/var/www/html/Pétalos_Studio/
```

Dentro de esta carpeta la estructura es:

```
Pétalos_Studio/
```

```
|— bootstrap/      # Archivos de Bootstrap ( css y javascript )  
|— img/           # Imágenes usadas en el proyecto  
|— inicio/        # Archivos principales frontend  
|   |— inicio.html  
|   |— inicio.css
```

2.5 Uso diario del entorno en VS Code



- Abrir **Visual Studio Code**.
- Ir al menú **Archivo** → **Abrir carpeta...** y seleccionar:
`/var/www/html/Pétalos_Studio/inicio`
- Hacer clic derecho sobre **inicio.html** y seleccionar **Open with Live Server**.
- Permitir que **Live Server** abra la página en el navegador y realice recarga automática al guardar.
- Iniciar **Live Share** para colaborar en tiempo real y compartir el enlace con los demás desarrolladores.

3. Instrucciones para replicar el entorno

1. Instalar Linux Mint 21.1 en VirtualBox o en máquina física.
2. Instalar Apache, PHP y MySQL con los comandos indicados.
3. Descargar e instalar Visual Studio Code.
4. Instalar solo las extensiones Live Server y Live Share en VS Code.
5. Crear la carpeta **Pétalos Studio** al escritorio del usuario, luego Copiar la carpeta del proyecto a `/var/www/html/Pétalos_Studio/` y dar permisos (ver 2.4).
6. Abrir el proyecto en VS Code y usar Live Server para desarrollo local.

Commits

Commits desde Git Bash utilizando “git log”



```
Kanoa Irrazabal@DESKTOP-T3D03V2 MINGW64 ~/Desktop/PetalosStudio (main)
$ git log
commit 4ac947edf6e2525e3a4cadfffffa7d56deb993178 (HEAD -> main, origin/main)
Author: Joaquín <joaquinirrazabal947@gmail.com>
Date: Sun Jul 13 22:11:47 2025 -0300

    Se agregaron los documentos

commit 5a9709f6abb2d7fd13b26b654ca03c371ff49e18
Author: Joaquín Irrazabal18 <joaquinirrazabal947@gmail.com>
Date: Sun Jul 13 21:38:29 2025 -0300

    Creamos el README.md

commit 58bf2d98899eba3d6ac5e7723d6aaf6015a8d075
Author: Joaquín <joaquinirrazabal947@gmail.com>
Date: Sat Jul 12 23:06:01 2025 -0300

    Colocamos el código original

commit 5bdacbf866800c9a737a40dfcaa7cc4b9eb8fd16
Author: Joaquín <joaquinirrazabal947@gmail.com>
Date: Sat Jul 12 22:23:33 2025 -0300

    Se agrega el Proyecto
```

Commits

main

Commits on Jul 13, 2025

- Se agregaron los documentos
Joaquinirrazabal18 committed 1 minute ago
4ac947e
- Creamos el README.md
Joaquinirrazabal18 authored 36 minutes ago
5a9709f (Verified)

Commits on Jul 12, 2025

- Colocamos el código original
Joaquinirrazabal18 committed yesterday
58bf2d9
- Se agrega el Proyecto
Joaquinirrazabal18 committed yesterday
5bdacbf

Commits desde el apartado de “Commits” desde GitHub

4. Modelado de Datos

4.1 Modelo Entidad-Relación (DER)

Para tener una mejor vista del DER, ir al Anexo

Entidades:

- Sistema Informático de Gestión de Partidas para Draft saurus IAE 3ºMA 13



6. Dino_ficha (**id_dino_ficha**, id_partida, especie, ubicacion_actual)
7. Dino_en_rio_descarte (**id_dino_en_rio_descarte**, id_dino_ficha, id_participacion, id_dado_turno, fecha_hora_descarte)
8. Restriccion (**id_restriccion**, nombre_restriccion (Único), descripcion)
9. Dado_turno (**id_dado_turno**, id_ronda, id_participacion, id_restriccion, numero_turno_en_ronda, resultado_dado, fecha_hora_lanzamiento)
10. Recinto (**id_recinto**, id_tablero, nombre_recinto, capacidad_maxima, area, lado)
11. Colocado (**id_colocado**, id_dino_ficha, id_participacion, id_recinto, id_dado_turno, posicion_dentro_recinto, fecha_hora_colocacion)
12. Tablero (**id_tablero**, tipo_tablero)

Identificación de Claves Foráneas (FKs)

{id_usuario} es FK de Participacion_p y referencia a {id_usuario} de Usuario

{id_partida} es FK de Participacion_p y referencia a {id_partida} de Partida

{id_tablero} es FK de Participacion_p y referencia a {id_tablero} de Tablero

{id_campeonato} es FK de Partida y referencia a {id_campeonato} de Campeonato

{id_partida} es FK de Ronda y referencia a {id_partida} de Partida

{id_partida} es FK de Dino_ficha y referencia a {id_partida} de Partida

{id_dino_ficha} es FK de Dino_en_rio_descarte y referencia a {id_dino_ficha} de Dino_ficha

{id_participacion} es FK de Dino_en_rio_descarte y referencia a {id_participacion_p} de Participacion_p

{id_dado_turno} es FK de Dino_en_rio_descarte y referencia a {id_dado_turno} de Dado_turno

{id_ronda} es FK de Dado_turno y referencia a {id_ronda} de Ronda

{id_participacion} es FK de Dado_turno y referencia a {id_participacion_p} de Participacion_p



{id_restriccion} es FK de Dado_turno y referencia a {id_restriccion} de Restriccion

{id_tablero} es FK de Recinto y referencia a {id_tablero} de Tablero

{id_dino_ficha} es FK de Colocado y referencia a {id_dino_ficha} de Dino_ficha

{id_participacion} es FK de Colocado y referencia a {id_participacion_p} de Participacion_p

{id_zona} es FK de Colocado y referencia a {id_recinto} de Recinto

{id_dado_turno} es FK de Colocado y referencia a {id_dado_turno} de Dado_turno

Modelo Entidad Relación (RNE):

$id^* > 0 \wedge id^* \in \text{Natural}$

- Aplica a: id_usuario, id_participacion_p, id_partida, id_campeonato, id_ronda, id_dino_ficha, id_dino_en_rio_descarte, id_restriccion, id_dado_turno, id_recinto, id_colocado, id_tablero.

$\text{Cantidad}^* > 0$

- Aplica a: num_participantes (en Partida), capacidad_maxima (en Recinto).

$\text{Cantidad}^* \geq 0$

- Aplica a: dinos_iniciales_bolsa (en Partida), dinos_restantes_bolsa (en Partida), ronda_actual (en Partida), turno_actual (en Partida), puntuacion_final (en Participacion_p).

$\text{Fecha_nacimiento} < \text{Fecha_registro}$

- Aplica a: fecha_nacimiento y fecha_registro en Usuario.

$\text{Fecha_registro} \leq \text{Ultimo_login}$

- Aplica a: fecha_registro y ultimo_login en Usuario.

$\text{Fecha_fin} > \text{Fecha_inicio}$

- Aplica a: fecha_hora_fin y fecha_hora_inicio en Ronda.

Orden Cronológico de Eventos en Partida:

- Partida->fecha_creacion es el inicio de todos los eventos relacionados.
- Participacion_p->fecha_hora_inicio \geq Partida->fecha_creacion.
- Ronda->fecha_hora_inicio debe ser secuencial y posterior a la anterior Ronda dentro de la misma Partida.
- Dado_turno->fecha_hora_lanzamiento debe ser secuencial dentro de una Ronda y posterior a Ronda->fecha_hora_inicio.
- Dino_en_rio_descarte->fecha_hora_descarte debe ser posterior o igual a Dado_turno->fecha_hora_lanzamiento al que está asociado.
- Colocado->fecha_hora_colocacion debe ser posterior o igual a Dado_turno->fecha_hora_lanzamiento al que está asociado.

Posición dentro de Recinto: Debe ser una coordenada o índice válido dentro del área del recinto correspondiente.



Enums:

- Usuario->rol = {"Jugador", "Administrador"}
- Participacion_p->estado_participacion = {"Activa", "Completada", "Abandono"}
- Partida->estado_partida = {"creada", "en_curso", "pausada", "finalizada", "cancelada"}
- Partida->lado_tablero_usado = {"Verano", "Invierno"}
- Dino_ficha->especie = {"Tiranosaurio Rex", "Triceratops", "Brachiosaurus", "Stegosaurus", "Pterodáctilo", "Plesiosaurio"}
- Tablero->tipo_tablero = {"Verano", "Invierno"}
- Recinto->lado = {"Baños", "Cafetería"}
- Recinto->area = {"Bosque", "Llanura"}
- Dado_turno->resultado_dado = {"El Bosque", "Llanura", "Baños", "Cafetería", "Recinto Vacío", "¡Cuidado con el T-Rex!"}
- Recinto->nombre_recinto = {"El Bosque de la Semejanza", "El Prado de la Diferencia", "La Pradera del Amor", "El Trío Frondoso", "El Rey de la Selva", "La Isla Solitaria", "El Bosque Ordenado", "El Puente de los Enamorados", "La Pirámide", "El Puesto de Observación", "Zona de Cuarentena"}

Repositorio en GitHub

<https://github.com/JoaquinIrrazabal18/PetalosStudio>