



UNIVERSIDAD
DON BOSCO

GOST

INTEGRANTES

-  LESLY MICHELLE MARTÍNEZ ROMERO MR222351
-  EDGAR JOSUÉ GÓMEZ MELÉNDEZ GM240279
-  EDGAR ANTONIO RUIZ HERNÁNDEZ RH201851

MATERIA: DESARROLLO DE APLIC. WEB CON SOFT. INTERPRET. EN EL SERVIDOR INGENIERO: RENÉ MAURICIO TEJADA

LINK DEL REPOSITORIO GIT: [HTTPS://GITHUB.COM/IMLESLY/HTTPS-GITHUB.COM-IMLESLY-INVESTIGACI-N-APLICADA-2_DSS.GIT](https://github.com/imlesly/https-github.com-imlesly-investigaci-n-aplicada-2_dss.git)

Definición del Paradigma Event-Driven:

Es un modelo de desarrollo en el que el flujo de ejecución de un programa es determinado por eventos en lugar de seguir una secuencia predefinida de instrucciones. Estos eventos pueden ser acciones del usuario, respuestas de servidores, cambios en el estado de una aplicación, entre otros.

Principios fundamentales del paradigma event-driven

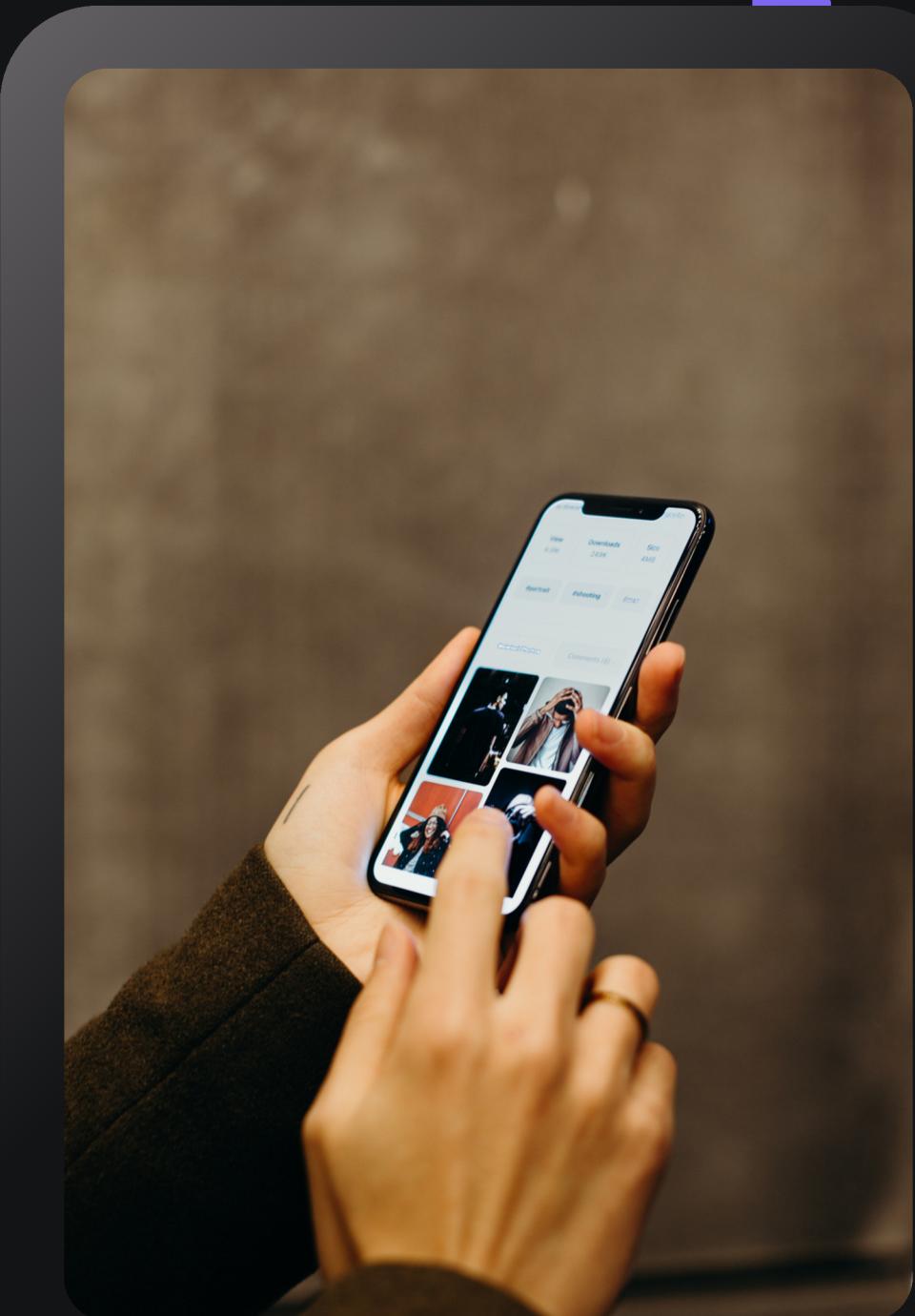
1 Eventos como desencadenadores de acciones

3 Desacoplamiento de componentes

2 Observadores y emisores de eventos

Aplicación en la gestión de solicitudes en servidores:

- El procesamiento asíncrono en tecnologías
- Webhook y eventos en tiempo real



Estado Actual de PHP

Modelo tradicional de ejecución de PHP:

Ejecución basada en procesos

Interacción con el servidor web:

Gestión de recursos



Limitaciones del modelo síncrono en PHP:

- Bloqueo de procesos
- Consumo elevado de recursos:
- Dificultad para manejar eventos en tiempo real

FRAMEWORKS Y LIBRERÍAS EVENT-DRIVEN EN PHP:

SWOOLE:	REACTPHP:
<ul style="list-style-type: none">Soporte para coroutines: Permite la ejecución de múltiples tareas de manera concurrente, optimizando el uso de recursos.Servidor HTTP, WebSockets y RPC: Facilita la creación de servidores asíncronos sin necesidad de Apache o Nginx.Gestión avanzada de procesos y tareas: Implementa pools de trabajadores para mejorar el rendimiento en sistemas con alta concurrencia.Soporte para microservicios: Permite la comunicación eficiente entre servicios mediante eventos.	<ul style="list-style-type: none">Event Loop: Un bucle de eventos que permite gestionar múltiples tareas de manera simultánea sin bloquear el proceso principal.Servidor HTTP asíncrono: Facilita la construcción de servidores sin necesidad de utilizar Apache o Nginx.Soporte para WebSockets y Streams: Permite la comunicación en tiempo real y la gestión de datos de manera asíncrona.Interacción con bases de datos y APIs: Optimiza la ejecución de consultas sin interrumpir el flujo de la aplicación.

Comparación entre Swoole y ReactPHP

Característica	Swoole	ReactPHP
Tipo	Extensión de PHP	Librería PHP
Modelo de ejecución	Asíncrono y basado en eventos	Asíncrono y basado en eventos
Soporte para coroutines	Sí	No
Servidor HTTP/WebSockets	Sí	Sí
Uso recomendado	Aplicaciones en tiempo real, servidores de alto rendimiento	APIs, servidores concurrentes

Ventajas y Desventajas de Cada Modelo

Aspecto	Thread-Based (Python)	Event-Driven (Node.js)
Rendimiento	Mayor latencia debido a la creación de hilos	Menor latencia gracias al event loop
Uso de CPU	Alto consumo en aplicaciones con muchas solicitudes	Optimizado para manejar múltiples conexiones
Gestión de Memoria	Mayor consumo de memoria por cada hilo	Uso eficiente de memoria al evitar múltiples hilos
Escalabilidad	Menos eficiente en cargas altas	Excelente para aplicaciones con miles de conexiones simultáneas
Complejidad de Desarrollo	Más sencillo de entender y depurar	Puede ser más complejo debido a la programación asíncrona

Comparación con Modelos Basados en Hilos

Aspecto	Event-Driven (PHP)	Thread-Based (Python, JavaScript)
Rendimiento	Menor latencia gracias al event loop	Mayor latencia por gestión de hilos
Escalabilidad	Manejo eficiente de múltiples conexiones	Puede requerir más recursos en alta concurrencia
Consumo de Memoria	Optimizado al evitar múltiples hilos	Mayor consumo por cada hilo creado
Complejidad de Desarrollo	Puede ser más complejo por la programación asíncrona	Más sencillo de entender y depurar

Desafíos y Consideraciones al Implementar un Enfoque Event-Driven en PHP:

- Compatibilidad con Entornos Tradicionales
- Curva de Aprendizaje de Arquitecturas Asíncronas
- Gestión de Recursos y Consumo de Memoria

Mejores Prácticas para la Implementación de Soluciones Event-Driven en PHP:

- **Elegir el Framework Adecuado**
- **Optimizar la Gestión de Eventos:**
- **Monitoreo y Depuración Avanzada**

Conclusiones y Futuras Direcciones del Paradigma Event-Driven en PHP:

1. Optimización de Procesos Asíncronos:

- Permite gestionar múltiples solicitudes simultáneamente sin bloquear la ejecución del sistema.
- Tecnologías como ReactPHP y Swoole han mejorado la capacidad de PHP para manejar eventos.

2. Comparación con Modelos Basados en Hilos

- El enfoque event-driven optimiza el uso de memoria y CPU en comparación con la ejecución tradicional basada en hilos.
- Es ideal para aplicaciones en tiempo real, como servidores de chat o plataformas de notificaciones.

3. Desafíos en la Implementación

- La compatibilidad con entornos tradicionales de PHP sigue siendo una barrera.
- La adopción de eventos en PHP requiere una curva de aprendizaje más pronunciada

Implicaciones en el Futuro del Desarrollo en PHP

- Mayor Adopción de Arquitecturas Asíncronas: El desarrollo de PHP está avanzando hacia la integración de sistemas asíncronos mediante extensiones como Swoole.

- **Expansión en el Uso de WebSockets y Streams:** Se espera un aumento en aplicaciones que requieren comunicación en tiempo real.
- **Mejoras en Seguridad y Estandarización:** A medida que PHP avanza en el manejo de eventos, surgirán prácticas más robustas para proteger los datos y optimizar la seguridad.

Futuras Líneas de Investigación

ü Integración de PHP con Tecnologías Serverless: Estudiar cómo PHP puede aprovechar plataformas serverless para mejorar la eficiencia en el manejo de eventos.

ü Comparación con Otros Lenguajes Asíncronos: Investigar cómo PHP se compara con JavaScript (Node.js) y Python en términos de rendimiento y concurrencia.

Uso de Event-Driven en Inteligencia Artificial: Explorar aplicaciones donde PHP y eventos puedan contribuir a procesamiento distribuido y aprendizaje automático.

Enlaces bibliográficos

- [1] "Manual oficial de eventos en PHP," PHP Manual.
Disponible en:
<https://www.php.net/manual/es/book.event.php>
- [2] "Cómo trabajar con eventos en PHP: ejemplos y pasos simples," Creación Web Profesional. Disponible en:
<https://creacionwebprofesional.org/php/como-trabajar-con-eventos-en-php-ejemplos-y-pasos-simples/>
- [3] "Modelo de trabajo de PHP," ProgrammerClick, [En línea].
Disponible en:
<https://programmerclick.com/article/27572672317/>
- [4] "Proceso de ejecución de PHP," ProgrammerClick, [En línea]. Disponible en:
<https://programmerclick.com/article/7084328112/>
- [5] "¿Ciclos PHP son síncronos o asíncronos?," Stack Overflow en español, [En línea]. Disponible en:
<https://es.stackoverflow.com/questions/465519/ciclos-php-son-sincronos-o-asincronos?citationMarker=43dcd9a7-70db-4a1f-b0ae-981daa162054>
- [6] "Uso de la interfaz síncrona y la interfaz asíncrona en PHP," ProgrammerClick, [En línea]. Disponible en:
<https://programmerclick.com/article/85741368792/>
- [7] "Documentación oficial de Swoole," OpenSwoole, [En línea]. Disponible en: <https://openswoole.com/docs>
- [8] "ReactPHP: Guía oficial," ReactPHP, [En línea]. Disponible en:
<https://reactphp.org/>
- [9] "Threads vs Events," CS5204 – Operating Systems, Virginia Tech, [En línea]. Disponible en: <https://courses.cs.vt.edu/cs5204/fall09-kafura/Presentations/Threads-VS-Events.pdf>
- [10] M. Cerón, "Complejidad de desarrollo en modelos basados en hilos," Facultad de Ciencias de la Computación, BUAP, [En línea]. Disponible en: https://www.cs.buap.mx/~mceron/cap2_dis.pdf
- [11] "Desarrollo de aplicaciones con arquitecturas event-driven," LovTechnology, [En línea]. Disponible en:
<https://lovtechnology.com/desarrollo-de-aplicaciones-con-arquitecturas-event-driven/>
- [12] "Arquitectura basada en eventos," IBM, [En línea]. Disponible en: <https://www.ibm.com/mx-es/topics/event-driven-architecture>
- [13] "Guía de Swoole en inglés," Swoole UK, [En línea]. Disponible en: <https://www.swoole.co.uk/>