

Investigación Aplicada 2: Implementación de Event-Driven en PHP

Instrucciones Generales:

- Esta actividad se llevará a cabo en grupos de 4 a 6 personas.
- Se realizará una exposición y presentación en video de la resolución del ejercicio en (máximo 10 diapositivas).
- Cada participante deberá defender su trabajo de manera individual.
- La presentación debe incluir los nombres y números de carnet de todos los miembros del grupo.
- También debe contener el enlace al repositorio de Git que crearon.
- La presentación se deberá subir en formato PDF al aula digital.
- La bibliografía debe incluir:
 - Documentos científicos publicados en revistas o congresos.
 - Documentos de instituciones de educación superior, priorizando los de la UDB: <https://rd.udb.edu.sv/home>.
 - Enlaces bibliográficos en formato IEEE.
- La entrega de la presentación se realizará a través del aula virtual.
- Se anulará la nota a ambos grupos en caso de que se presenten trabajos similares.
- Todas las consultas serán atendidas a través de estos canales de comunicación, en un horario de 8:00 am a 5:00 pm, con un retraso máximo de un día:
 - Chat AulaDigital.
 - Telegram

FECHA DE ENTREGA: 11 de Abril

Importante:

- *Cada punto no presentado cómo se solicita en las indicaciones generales, se le restará 0.20 puntos a su calificación total de la actividad.*
- *Quien no asistan a la exposición no tendrán derecho a calificación en la actividad y se le pondrá como nota 0.*

Desarrollo de Trabajo:

Investigar y aplicar técnicas para escalar aplicaciones web event-driven desarrolladas con lenguajes interpretados en el servidor, utilizando estrategias de horizontalización (replicación de instancias), balanceadores de carga, contenedores (Docker) y orquestadores (Kubernetes)

Parte I: Elaborar exposición sobre la implementación del paradigma Event-Driven en PHP.

Se deben centrar en los siguientes puntos:

1. Definición del Paradigma Event-Driven:

- Explicar el concepto de programación basada en eventos y su importancia en el desarrollo de aplicaciones web.
- Analizar los principios fundamentales del paradigma event-driven y cómo estos se aplican en el contexto de la gestión de solicitudes en servidores.

2. Estado Actual de PHP:

- Investigar el modelo tradicional de ejecución de PHP, centrado en la ejecución síncrona y la gestión de procesos.
- Identificar las limitaciones de este enfoque en el manejo de altas cargas de solicitudes simultáneas.

3. Frameworks y Librerías Event-Driven en PHP:

- Examinar frameworks y librerías como Swoole y ReactPHP, que permiten a PHP adoptar un enfoque event-driven.
- Detallar las características y funcionalidades que ofrecen estas herramientas para la gestión de eventos, la programación asíncrona y la comunicación en tiempo real.

4. Comparación con Modelos Basados en Hilos:

- Investigar el enfoque thread-based utilizado en otros lenguajes interpretados en el servidor, como Python (usando frameworks como Flask o Django) y JavaScript (Node.js).
- Analizar las ventajas y desventajas de cada enfoque, enfocándose en:

- **Rendimiento:** Comparar la latencia, el uso de CPU y la eficiencia en la gestión de memoria.
 - **Escalabilidad:** Evaluar la capacidad de cada modelo para manejar incrementos en la carga de trabajo.
 - **Complejidad de Desarrollo:** Examinar cómo cada paradigma afecta el proceso de desarrollo, incluyendo el manejo de errores y la complejidad del código.
5. **Casos de Uso y Aplicaciones Prácticas:**
- Proporcionar ejemplos de aplicaciones reales que utilizan un enfoque event-driven en PHP y cómo han beneficiado a estas implementaciones.
 - Comparar estos casos con aplicaciones similares en modelos basados en hilos, destacando diferencias en rendimiento y escalabilidad.
6. **Desafíos y Consideraciones:**
- Identificar los principales desafíos al implementar un enfoque event-driven en PHP, incluyendo cuestiones de compatibilidad y el aprendizaje de nuevas arquitecturas.
 - Discutir las mejores prácticas para la implementación de soluciones event-driven en PHP, así como consideraciones de seguridad.
7. **Conclusiones y Futuras Direcciones:**
- Resumir los hallazgos de la investigación y discutir las implicaciones del paradigma event-driven en el futuro del desarrollo en PHP.
 - Proponer futuras líneas de investigación en el área de programación basada en eventos y su impacto en la evolución de PHP y otros lenguajes de programación.

Preparación de la Exposición:

1. Investigar cada uno de los puntos mencionados para comprender a fondo los conceptos y herramientas.
2. Diseñar una presentación visualmente atractiva y clara que incluya toda la información relevante.

3. Preparar una presentación oral apoyada en medios visuales (canvas, PowerPoint, etc.), asegurando que se abordan todos los puntos de manera coherente y estructurada.
4. Practicar la exposición para garantizar que la duración de la presentación sea adecuada y que cada punto se explique con claridad.

Duración: La exposición debe tener una duración aproximada de 10 minutos, seguida de una sesión de preguntas y respuestas.

Parte II: Desarrollo de un sitio web dinámico con un servidor web no bloqueante en PHP utilizando el paradigma Event-Driven con ReactPHP.

Definición del problema: Una empresa X necesita un sitio web dinámico utilizando ReactPHP para manejar múltiples solicitudes simultáneas de manera no bloqueante. El sitio web tendrá varias páginas que responden a diferentes tipos de solicitudes (páginas estáticas, lectura de archivos, consultas a bases de datos) y todo será gestionado a través del servidor web basado en eventos.

Requerimientos de software:

1. Servidor Event-Driven:
 - Framework: Utilizar ReactPHP para implementar un servidor web basado en eventos que maneje múltiples solicitudes concurrentes sin bloquear el proceso principal.
 - Rutas básicas:
 - Página de Inicio ("/"): Cargar una página HTML estática que sirva como punto de entrada al sitio.
 - Página de Contacto ("/contact"): Cargar una página con un formulario para enviar información de contacto.
 - Ruta de Datos ("/data"): Cargar datos dinámicos de una fuente (como un archivo o base de datos) y presentarlos en formato JSON o HTML.
 - Archivos estáticos ("/style.css"): Servir archivos CSS y recursos estáticos (como imágenes) sin bloquear otras solicitudes.

2. Manejo Asíncrono de Solicitudes:

- Asincronía en consultas y archivos: Realizar consultas a archivos o bases de datos de manera no bloqueante utilizando promesas o flujos asíncronos, para evitar que el servidor se detenga o bloquee al manejar múltiples solicitudes.
- Control de errores: Implementar manejo adecuado de errores, como rutas no encontradas (404) o fallos en la carga de archivos y datos (500).

3. Simultaneidad:

- El servidor debe poder gestionar múltiples solicitudes simultáneas (por ejemplo, varios usuarios accediendo al sitio o solicitando datos) de forma eficiente y sin interrupciones. Esto será posible gracias al paradigma event-driven, que evitará que las operaciones de entrada/salida bloqueen el flujo de trabajo del servidor.

4. Operaciones CRUD:

- Implementar operaciones básicas de consulta y manejo de datos (Create, Read, Update, Delete) en la página /data. Por ejemplo, permite agregar, actualizar o eliminar entradas en una base de datos, y todo esto debe ser gestionado de manera asíncrona.

Desarrollo de Trabajo:

Actividad	Criterio a Evaluar	Puntuación
Exposición (30%)	Claridad y Precisión de la Información: La información es precisa, clara y completa. Se explican todos los puntos con detalle y exactitud.	6
	Estructura y Organización: La exposición está muy bien organizada, con una introducción clara, desarrollo coherente y conclusión efectiva.	6

	Calidad Visual de la Presentación: La infografía es visualmente atractiva, fácil de entender y bien diseñada. Usa gráficos e imágenes de manera efectiva.	6
	Compresibilidad de la presentación: La presentación presenta la información de manera comprensible, con uso adecuado de texto y gráficos	6
	Manejo del Tiempo: La exposición se ajusta perfectamente al tiempo asignado, cubriendo todos los puntos sin apresurarse ni extenderse.	6
**Ejercicios (50%)	Implementación del servidor ReactPHP <ul style="list-style-type: none"> El servidor está correctamente configurado utilizando ReactPHP y maneja solicitudes simultáneas de manera no bloqueante. 	5
	Uso del modelo Event-Driven <ul style="list-style-type: none"> Verificar que el paradigma basado en eventos se utiliza correctamente, con el loop de eventos (EventLoop) y "event listeners" para gestionar solicitudes y respuestas 	5

	<p>Rutas y páginas correctamente implementadas</p> <ul style="list-style-type: none"> Las rutas deben estar correctamente configuradas. Cada página (estática o dinámica) debe responder de acuerdo con la solicitud (incluyendo la navegación entre páginas). 	5
	<p>Operaciones asíncronas con base de datos</p> <ul style="list-style-type: none"> Las consultas SQL deben ser gestionadas de manera asíncrona, sin bloquear el servidor. Se evalúa que las operaciones de lectura y escritura en la base de datos sean eficientes. 	5
	<p>Manejo de solicitudes concurrentes</p> <ul style="list-style-type: none"> El servidor debe ser capaz de procesar varias solicitudes simultáneas sin bloquear, demostrando eficiencia bajo carga (pruebas de estrés o simulación de múltiples usuarios) 	5
	<p>Validación y seguridad</p> <ul style="list-style-type: none"> Implementar validaciones de entradas de usuarios (formularios) y mecanismos de seguridad contra ataques comunes (inyección SQL, XSS) 	5

	Funcionalidad <ul style="list-style-type: none"> • Verificar que las rutas estén correctamente definidas y las páginas respondan adecuadamente a las solicitudes, gestionando correctamente los errores. • Asegurar que las operaciones de lectura y escritura en la base de datos se gestionen de manera no bloqueante utilizando mecanismos asincrónicos. 	20
VIDEO (20%)	Video donde presenta exposicion y aplicacion	20

***Las secciones marcadas con asterisco solo serán calificadas si todos los ejercicios solicitados están en un repositorio funcional. Si no se cumplen estas condiciones, la sección recibirá automáticamente una calificación de cero.*