

Investigación y Aplicación de una Arquitectura de Programación en un Sistema de Información

Daniel Gil Castillo
Profesor: Carlos Basulto
Asignatura: Desarrollo Web en Entorno Servidor



Índice:

Índice:	2
1. Investigación de Arquitecturas: Los estudiantes deben investigar sobre al menos tres arquitecturas de programación. Algunas de las más comunes incluyen:.....	3
2. Selección de Arquitectura: El estudiante deberá elegir una arquitectura que se adapte a las necesidades del sistema de información a desarrollar. Justificar la elección de la arquitectura seleccionada, basándose en las ventajas y desventajas de cada una.....	3
3. Caso de Estudio (Sistema de Información/Aplicación): Proponer un caso de estudio donde se implementará el sistema. El sistema puede ser de cualquier tipo (gestión de inventarios, sistema de reservas de hoteles, plataforma educativa, etc.).....	4
○ Descripción general del sistema.....	4
○ Identificación de las necesidades del sistema (funcionales y no funcionales).....	4
1. Requisitos funcionales:.....	4
2. Requisitos no funcionales:.....	5
4. Diagrama de Arquitectura: Crear un diagrama que represente la arquitectura seleccionada para el sistema. El diagrama debe incluir todos los componentes principales del sistema (servidores, bases de datos, servicios, etc.) y las interacciones entre ellos.....	6
5. Documentación: Entregar un informe detallado que incluya:.....	6
● Introducción al sistema y su caso de estudio.....	6
● Explicación detallada de la arquitectura elegida.....	6
● Diagrama de arquitectura.....	6
● Justificación de las decisiones tomadas durante el desarrollo del sistema.....	6
● Descripción de cómo se implementaría la arquitectura en la práctica, incluyendo herramientas y lenguajes de programación sugeridos.....	6
Introducción al sistema y su caso de estudio.....	6
Explicación Detallada de la Arquitectura Elegida.....	7
Diagrama de Arquitectura.....	7
Justificación de las Decisiones Tomadas durante el Desarrollo del Sistema.....	8
Descripción de la Implementación Práctica de la Arquitectura.....	8
Referencias:	9



Unión Europea

Daniel Gil Castillo
Profesor: Carlos Basulto
Asignatura: Desarrollo Web en Entorno Servidor

1. Investigación de Arquitecturas: Los estudiantes deben investigar sobre al menos tres arquitecturas de programación. Algunas de las más comunes incluyen:

Capa de Presentación (Presentation Layer): interfaz de usuario, presenta la información al usuario y recoge la entrada del usuario (interfaces gráficas de usuario (GUI), páginas web o servicios web).

Capa de Lógica de Aplicación (Business Logic Layer): reglas específicas de la aplicación: procesamiento de datos y toma de decisiones basada en la lógica del negocio. No debe depender de detalles de la capa de acceso a datos.

Capa de Acceso a Datos (Data Access Layer): Interactúa con la fuente de datos (base de datos, servicios web y otro tipo de almacenamiento). Proporciona interfaz para que la capa de lógica de aplicación acceda a los datos.

2. Selección de Arquitectura: El estudiante deberá elegir una arquitectura que se adapte a las necesidades del sistema de información a desarrollar. Justificar la elección de la arquitectura seleccionada, basándose en las ventajas y desventajas de cada una.

He escogido el sistema de información de una Tienda Online, en este caso, debido a mi interés por desarrollar la aplicación en el futuro, quiero adecuarla a nuevas funcionalidades ,y hacerla más extensible y debido a la necesidad de la mayor



Unión Europea

Daniel Gil Castillo
Profesor: Carlos Basulto
Asignatura: Desarrollo Web en Entorno Servidor

escalabilidad posible, escoger la arquitectura de microservicios al permitirme incrementar sus funciones en el futuro.

3. Caso de Estudio (Sistema de Información/Aplicación):
Proponer un caso de estudio donde se implementará el sistema. El sistema puede ser de cualquier tipo (gestión de inventarios, sistema de reservas de hoteles, plataforma educativa, etc.).

Debe incluir los siguientes puntos:

- **Descripción general del sistema.**

Mi sistema se trata de una Tienda Online, la cuál incluye una página de login, a la cuál se accede con un usuario y contraseña, un catálogo, distribuido en familias de tipos de productos, y un carrito con una plataforma de pago.

- **Identificación de las necesidades del sistema (funcionales y no funcionales).**

1. Requisitos funcionales:

Servicios a proporcionar:

- 1. Login de usuario:**
 - Autenticación mediante usuario y contraseña.
 - Recuperación de contraseña mediante correo electrónico.
 - Registro de nuevos usuarios.
- 2. Gestión de usuarios:**
 - Creación, edición y eliminación de cuentas de usuario.
- 3. Catálogo de productos:**
 - Visualización de productos organizados en familias de tipos.
 - Filtros y búsqueda por categorías, precio, popularidad o novedades.
 - Páginas de detalles para cada producto con imágenes y descripciones.
- 4. Carrito de compras:**
 - Adición, modificación y eliminación de productos en el carrito.
 - Visualización del total acumulado en tiempo real.
- 5. Plataforma de pago:**



Unión Europea

Daniel Gil Castillo
Profesor: Carlos Basulto
Asignatura: Desarrollo Web en Entorno Servidor

- Gestión segura de la información de pago.

2. Requisitos no funcionales:

Aspectos técnicos del sistema:

1. **Escalabilidad:**
 - Capacidad para manejar picos de tráfico durante promociones o temporadas altas.
2. **Seguridad:**
 - Cifrado de contraseñas y datos sensibles.
3. **Rendimiento:**
 - Respuesta rápida a las consultas del catálogo.
4. **Disponibilidad:**
 - Mecanismos de recuperación ante fallos.
5. **Mantenibilidad:**
 - Arquitectura modular que facilite la actualización de componentes individuales.
6. **Compatibilidad:**
 - Diseño responsive para funcionar en navegadores web y dispositivos móviles.

- Justificación de la elección de la arquitectura para ese sistema en particular.

La elección de una arquitectura de microservicios se justifica principalmente por las siguientes razones:

1. **Escalabilidad horizontal:**
 - Permite escalar servicios específicos de manera independiente.
2. **Desarrollo y despliegue independientes:**
 - Cada funcionalidad principal (autenticación, catálogo, carrito, pagos) se implementa como un servicio autónomo. Esto facilita que equipos diferentes trabajen en paralelo y acelera el desarrollo.
3. **Flexibilidad tecnológica:**
 - Cada microservicio puede ser desarrollado con el lenguaje y las herramientas más adecuadas para su propósito.
4. **Resiliencia:**
 - En una arquitectura monolítica, un fallo en un componente puede interrumpir todo el sistema. En microservicios, un error en el módulo de pagos, por ejemplo, no afecta al catálogo ni al login.
5. **Facilidad de integración de terceros:**

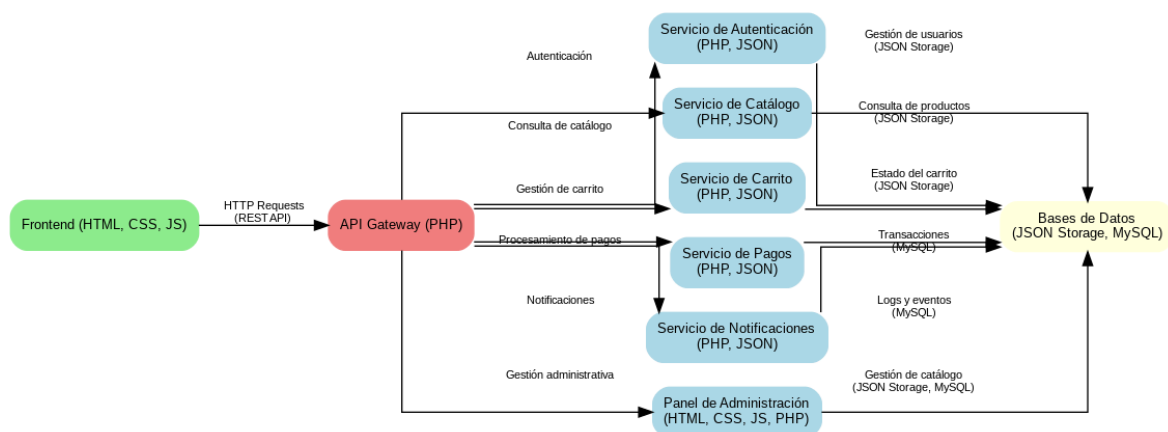


Unión Europea

Daniel Gil Castillo
Profesor: Carlos Basulto
Asignatura: Desarrollo Web en Entorno Servidor

- La modularidad permite integrar con APIs de terceros (como pasarelas de pago o sistemas de marketing) sin grandes cambios en el sistema.

4. Diagrama de Arquitectura: Crear un diagrama que represente la arquitectura seleccionada para el sistema. El diagrama debe incluir todos los componentes principales del sistema (servidores, bases de datos, servicios, etc.) y las interacciones entre ellos.



5. Documentación: Entregar un informe detallado que incluya:

- Introducción al sistema y su caso de estudio.
- Explicación detallada de la arquitectura elegida.
- Diagrama de arquitectura.
- Justificación de las decisiones tomadas durante el desarrollo del sistema.
- Descripción de cómo se implementaría la arquitectura en la práctica, incluyendo herramientas y lenguajes de programación sugeridos.



Unión Europea

Daniel Gil Castillo
 Profesor: Carlos Basulto
 Asignatura: Desarrollo Web en Entorno Servidor

Introducción al sistema y su caso de estudio.

El sistema de información desarrollado se centra en una tienda online que permite a los usuarios comprar productos de manera eficiente y segura. Este sistema incluye una variedad de funcionalidades esenciales, como una página de inicio de sesión (login), un catálogo de productos organizado en familias de tipos, un carrito de compras y una plataforma de pago. El objetivo principal del sistema es ofrecer una experiencia de usuario fluida mientras se garantiza la escalabilidad, la seguridad y el rendimiento.

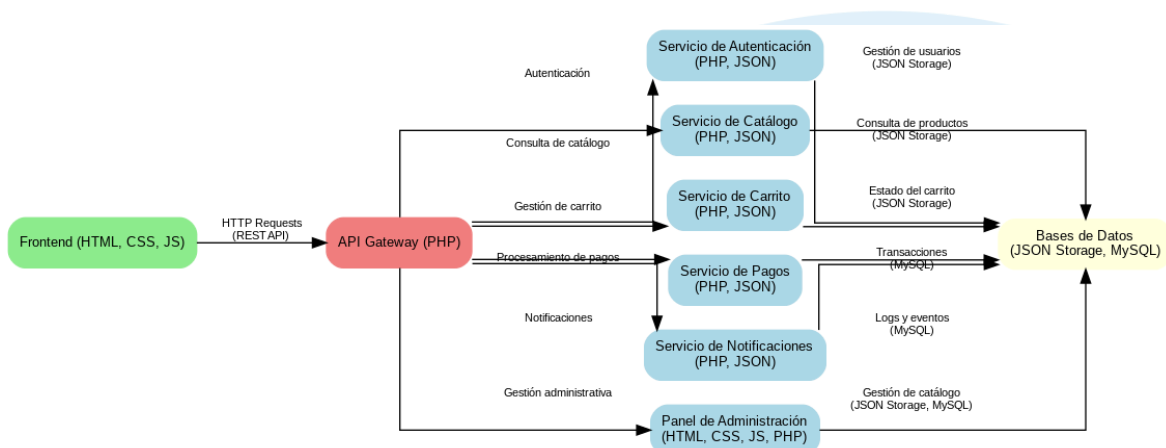
Utilizando una arquitectura basada en microservicios implementada con lenguajes y herramientas compatibles con HTML, CSS, JavaScript, JSON, PHP y MySQL, se logra un diseño modular, eficiente y escalable.

Explicación Detallada de la Arquitectura Elegida

Se ha seleccionado una arquitectura basada en microservicios debido a los beneficios que ofrece en términos de escalabilidad, resiliencia y mantenibilidad. Los principales componentes del sistema se dividen en servicios independientes que interactúan a través de un API Gateway desarrollado en PHP.

Cada microservicio es responsable de una funcionalidad específica del sistema, lo que permite que los equipos de desarrollo trabajen de manera paralela y que las actualizaciones o escalamiento se realicen de manera granular.

Diagrama de Arquitectura



Unión Europea

Daniel Gil Castillo
Profesor: Carlos Basulto
Asignatura: Desarrollo Web en Entorno Servidor

Justificación de las Decisiones Tomadas durante el Desarrollo del Sistema

1. **Arquitectura basada en microservicios:**

- Escalabilidad.
- Resiliencia.
- Flexibilidad.

2. **Tecnologías utilizadas:**

- **Frontend:** HTML, CSS y JavaScript son los estándares de desarrollo web y garantizan una experiencia de usuario atractiva y funcional.
- **Backend:** PHP es un lenguaje robusto y ampliamente utilizado para el desarrollo de APIs y servicios backend.
- **Bases de datos:** MySQL es una solución relacional que se adapta bien a las necesidades de la tienda, mientras que JSON se utiliza para el intercambio de datos entre servicios.

3. **Seguridad:**

- Implementación de autenticación segura para proteger los datos de los usuarios.

Descripción de la Implementación Práctica de la Arquitectura

La arquitectura se implementa de la siguiente manera utilizando los lenguajes especificados:

1. **Frontend:**

- **HTML y CSS:** Estructura y estilo de las interfaces de usuario, incluyendo páginas de login, catálogo y carrito de compras.
- **JavaScript:** Manejo de la interactividad en el navegador, como el añadido de productos al carrito y la actualización dinámica de precios.

2. **Backend:**

- **PHP (API Gateway):** Punto central para redirigir solicitudes a los microservicios correspondientes. Maneja la comunicación entre el frontend y los servicios backend.
- **Servicios individuales en PHP:**
 - Servicio de autenticación: Valida usuarios mediante JSON y almacena información en MySQL.



Unión Europea

Daniel Gil Castillo
Profesor: Carlos Basulto
Asignatura: Desarrollo Web en Entorno Servidor

- Servicio de catálogo: Proporciona datos de productos organizados, recuperados desde MySQL y servidos como JSON.
- Servicio de carrito: Administra el estado del carrito del usuario mediante JSON y sincronización con la base de datos.
- Servicio de pagos: Procesa transacciones seguras y las registra en MySQL.
- Servicio de notificaciones: Envía correos y alertas al usuario después de realizar pedidos.

3. Bases de Datos:

- **MySQL:** Gestiona datos estructurados como usuarios, productos, pedidos y transacciones.
- **JSON:** Se utiliza como formato de intercambio de datos entre microservicios para mantener la interoperabilidad.

4. Despliegue:

- Los microservicios se despliegan en servidores independientes, configurados para comunicarse mediante el API Gateway.
- El frontend se sirve desde un servidor web, Apache.

Referencias:

https://drive.google.com/file/d/1lq1iWtiWRflejeMeJAqskUqKoxoCUaJu/view?usp=classroom_web&authuser=0: Documento proporcionado para la realización de este proyecto.



Unión Europea

Daniel Gil Castillo
 Profesor: Carlos Basulto
 Asignatura: Desarrollo Web en Entorno Servidor