


Tarea

L2 | Ejercicios de aplicación #2 Arquitectura de software

Requisitos de finalización

Por hacer: Hacer un envío

 En esta sección podrás entregar la resolución del ejercicio propuesto. Para realizarlo de forma efectiva sigue los siguientes pasos:

- **Formato del archivo:** PDF google slide.
- **Nombre del archivo:** Apellido_Nombre | [L] N° lección: nombre de la lección ejercitada | [M] N° del módulo.
 - Ejemplo: *García Maria | L3: Introducción a la computación en la Nube | M3*
- **Envío del archivo:**
 - Puedes descargar localmente en tu computadora, el archivo donde desarrollaste el ejercicio para luego cargarlo en mediante el botón de "agregar entrega".
 - Crea una carpeta en Google Drive, sube tu archivo a esa carpeta y verifica que los permisos estén configurados como "Cualquiera con el enlace puede ver". Finalmente copia el enlace de la carpeta y pégalo en el campo habilitado para la entrega.

¡Mucho éxito!

Estado de la entrega

Estado de la entrega	Todavía no se han realizado envíos
Estado de la calificación	Sin calificar

Última modificación	-
Comentarios de la entrega	Comentarios (0)

Criterios de calificación

Este ejercicio no cuenta con resolución disponible.

Rúbrica				
Cumplimiento de la consigna	No entrega 0 puntos	Insuficiente 1 puntos	Adecuado 2 puntos	
Claridad y profundidad de la explicación	No entrega 0 puntos	Insuficiente 1 puntos	Adecuado 2 puntos	
Aplicación de conceptos clave	No entrega 0 puntos	Insuficiente 1 puntos	Adecuado 2 puntos	
Análisis y justificación	No entrega 0 puntos	Insuficiente 1 puntos	Adecuado 2 puntos	
Propuesta adicional o tarea complementaria	No entrega 0 puntos	Insuficiente 1 puntos	Adecuado 2 puntos	
Presentación y organización	No entrega 0 puntos	Insuficiente 1 puntos	Adecuado 2 puntos	

Ejercicios de aplicación

Arquitectura de software

Es hora de que pongas en práctica todo lo aprendido. 

Este apartado tiene el objetivo de ayudarte a seguir potenciando tus habilidades, por lo que a continuación encontrarás diferentes **desafíos** que podrás resolver de forma independiente y a tu ritmo.

¡Manos a la obra!

1. Desafío

Consigna: Define escenarios de calidad específicos para un sistema de gestión de reservas en línea. Elige atributos de calidad clave, como seguridad, rendimiento y usabilidad, y crea tres escenarios de calidad para cada atributo, especificando estímulo, contexto, respuesta esperada y métrica de respuesta. Luego, diseña un diagrama básico de la arquitectura que soporte estos atributos de calidad.

Pasos:

1. Selecciona tres atributos de calidad prioritarios para el sistema y describe su relevancia.
2. Define tres escenarios de calidad para cada atributo, detallando estímulo, contexto, respuesta esperada y métrica.
3. Crea un diagrama arquitectónico básico, usando herramientas de diseño, que represente cómo el sistema soportará estos escenarios de calidad.

4. ¿Dónde se lleva a cabo?

Puedes realizar este desafío en una herramienta de diagramación como **Lucidchart** o **Draw.io** para crear el diseño arquitectónico. Para documentar

los atributos de calidad y escenarios, puedes usar **Word**, **Google Docs** o **Notion**.

5. Tiempo de dedicación ⌚ 2 Horas.

6. Recursos

- **Manual del módulo** para repasar conceptos sobre atributos y escenarios de calidad.
- Referencia a la **ISO/IEC 25010**, que define los principales atributos de calidad en software, para asegurar una selección fundamentada de atributos.
- Guías en línea de diseño de escenarios de calidad, que ofrecen ejemplos de cómo detallar cada elemento de los escenarios (estímulo, contexto, respuesta esperada y métrica).

7. Plus

Opcional: Para profundizar en el desafío, agrega un análisis de riesgos para cada escenario de calidad, señalando los puntos críticos y proponiendo soluciones de mitigación. Además, puedes investigar cómo marcos de calidad como **AWS Well-Architected Framework** pueden fortalecer la arquitectura para cumplir con estos escenarios de calidad.

REPORTE

1. Selecciona 3 atributos de calidad prioritarios para el sistema y describe su relevancia.

- i) Seguridad, debido a que es clave efectuar una compra segura protegiendo la transacción y datos del cliente.
- ii) Usabilidad, el sistema debe ser de fácil uso para el cliente, que su interacción minimice errores de ingreso de datos.
- iii) Mantenibilidad, para la corrección de errores y facilitar la actualización. Diseñado de forma modular para facilitar constantes mejoras y ajustes.

2. Define tres escenarios de calidad para cada atributo, detallando estímulo, contexto, respuesta esperada y métrica

A continuación, se detallan tres escenarios para cada atributo de calidad, siguiendo la estructura de estímulo, contexto, respuesta y métrica.

Seguridad (Compra Segura)

	Escenario 1: Transacción de pago seguro (S1)	Escenario 2: Control de Acceso (S2)	Escenario 3: Intento de Inyección de datos (S3)
Estímulo	Un cliente realiza el pago de una reserva con su tarjeta de crédito/débito.	Un usuario no autenticado intenta acceder a una URL interna del sitio.	Un usuario envía código en un formulario de entrada
Contexto	El cliente está en el último paso del proceso de reserva.	El sistema se encuentra en operación normal.	El sistema se encuentra en operación normal.
Respuesta Esperada	La conexión se realiza bajo HTTPS. Los datos de la tarjeta son enviados a una pasarela de pago segura y no se almacenan en el sistema.	El sistema niega el acceso y redirige al usuario a la página de inicio de sesión.	El sistema valida y limpia la entrada, rechazando la petición.
Métrica	El 100% de las transacciones usan TLS 1.3.	El 100% de los accesos a rutas protegidas son denegados a usuarios anónimos.	100% de los intentos de inyección son bloqueados en pruebas de penetración.

Usabilidad (Fácil uso y acceso)

	Escenario 1: Manejo de Errores (U1)	Escenario 2: Acceso multiplataforma (U2)	Escenario 3: Proceso de Reserva Intuitivo (U3)
Estímulo	Un usuario introduce datos inválidos (ej: una fecha incorrecta) en el formulario de reserva.	Un usuario ingresa por móvil, Tablet o PC al sistema.	Un usuario nuevo intenta completar una reserva por primera vez.
Contexto	El usuario llena el formulario de búsqueda de reservas.	El usuario se encuentra en la página de inicio.	El usuario se encuentra en la página de inicio.
Respuesta Esperada	El sistema muestra un mensaje de error claro y en contexto, explicando cómo corregir el dato, sin recargar la página.	El usuario completa sin problemas la reserva.	El usuario es capaz de completar la reserva en pocos pasos, sin necesidad de ayuda externa.
Métrica	El 100% de los errores de validación de entrada muestran un mensaje útil en menos de 500ms.	El 100% de los usuarios de prueba completan la reserva en los 3 tipos de dispositivos soportados.	El 95% de los usuarios de prueba completan la reserva en menos de 3 minutos.

Mantenibilidad (Funcionar sin fallas)

	Escenario 1: Reserva Concurrente (M1)	Escenario 2: Aumento de Tráfico Inesperado (M2)	Escenario 3: Falla en la Base de Datos (M3)
Estímulo	Varios usuarios intentan reservar el último cupo disponible exactamente al mismo tiempo.	El número de usuarios concurrentes se quintuplica durante un evento de alta demanda (grandes descuentos)	El servidor de la base de datos principal deja de funcionar.
Contexto	El sistema procesa transacciones de forma concurrente.	El sistema está operando bajo alta carga.	El sistema está operando bajo carga normal.
Respuesta Esperada	Asegurar que solo un usuario complete la reserva. El otro recibe un mensaje de "cupos no está disponible, intente otra fecha".	El sistema escala horizontalmente, añadiendo más servidores de aplicación.	El sistema automáticamente se enlaza a una base de datos ubicado en otro site (balanceo de carga).
Métrica	0% de sobreventa de cupos en pruebas de estrés con alta concurrencia.	La tasa de errores de transacción es $< 0.1\%$.	La disponibilidad del sistema es del 99.95%.

3 Crea un diagrama arquitectónico básico, usando herramientas de diseño, que represente cómo el sistema soportará estos escenarios de calidad.

A continuación se describen los conceptos que contiene la arquitectura básica diseñada para representar los escenarios de calidad.

Descripción de Componentes y su Relación con la Calidad:

1. **Frontend App:** Corresponde a la aplicación que se ejecuta en el navegador del usuario. Es responsable directa de la **Usabilidad**, asegurando que la interfaz responda inmediatamente sobre errores (U1), multiplataforma (U2) y que sea de fácil uso (U3).
2. **API Gateway:** Es el único punto de entrada al sistema. Para la **Seguridad** centraliza el cifrado de las comunicaciones (S2). Puede contener componente Web Application Firewall (WAF) que bloquea ataques comunes como la Inyección SQL (S3).
3. **Servidor de Aplicaciones:** Contiene la lógica del negocio. Está diseñado como un clúster para soportar la **Mantenibilidad**, permitiendo el escalado horizontal (M2). Implementa la lógica de control de acceso para **Seguridad** (S2) y se comunica de forma segura con la pasarela de pagos.
4. **Balanceador de carga:** Para la **Mantenibilidad** se encarga de distribuir el tráfico entre múltiples servidores durante cuando existe alta demanda (M2).
5. **Base de Datos:** En la **Mantenibilidad** su configuración de réplica en otro site (activo-pasivo) permite la conmutación automática en caso de fallos (M3), gracias a sus propiedades transaccionales (ACID) previenen la corrupción de datos por reservas concurrentes (M1).
6. **Portal de Pagos:** Componente de servicio externo que robustece la **Seguridad** al procesar los datos de pago, evitando que sean almacenados en el sistema de reservas (S1).
7. **Servicio de Monitoreo:** Útil para la **Mantenibilidad** y **Seguridad**. Alerta sobre fallos (M3) , registra el estado del sistema, e informa sobre intentos de ataque (S3).

En la siguiente imagen, se ilustra un diagrama arquitectónico básico que grafica los conceptos descritos anteriormente referente a los escenarios de calidad.

