

UNIDAD TEMÁTICA 1 – UML – Trabajo de Aplicación 1

EJERCICIO 1

1) Plataforma de comercio electrónico:

Se necesitan representar las diferentes entidades y sus relaciones, como productos, clientes, carritos de compra y pedidos. El objetivo es la comprensión de la estructura general del sistema y la identificación de las responsabilidades de cada entidad.

DIAGRAMA DE CLASES - Quiere representar las entidades y sus relaciones, así como también la estructura general y las responsabilidades de cada entidad, estas son las operaciones y restricciones.

2) Creación de un sistema de autenticación y autorización:

Se quiere mostrar la interacción entre diferentes objetos y componentes durante el proceso de inicio de sesión, registro y asignación de roles/permisos. Esto permitiría a los desarrolladores entender el flujo de información y eventos en el sistema.

DIAGRAMA DE ACTIVIDAD - muestran una secuencia de acciones, un flujo de trabajo que va desde un punto inicial hasta un punto final. Permite comprender el flujo de funcionalidades.

3) Implementación de un proceso de negocio en una empresa:

Se quiere representar los diferentes pasos y decisiones involucradas en un proceso empresarial, como la aprobación de una factura o la contratación de un empleado. Esto ayudaría a los stakeholders a visualizar y optimizar el flujo de trabajo.

DIAGRAMA DE ACTIVIDAD - Este tipo de diagrama ofrece una visión a alto nivel de procesos del negocio y requisitos funcionales, al ser alto nivel, los stakeholders (pueden ser personas sin mucho conocimiento de la tecnología) pueden visualizar y entender más fácilmente.

4) Desarrollo de una aplicación móvil con múltiples módulos:

Se necesita visualizar los módulos y sus dependencias en la aplicación, como la interfaz de usuario, lógica de negocio y servicios de datos. Esto facilita la comprensión de la arquitectura general y la gestión de dependencias.

DIAGRAMA DE COMPONENTES - Habla de dependencia y módulos, este diagrama sirve para documentar las relaciones entre el sistema a través de las interfaces.

5) Creación de un sistema de gestión de contenido (CMS):

Se requiere describir la estructura y las interacciones entre los componentes del CMS, como el editor de texto, el gestor de archivos y el motor de plantillas. Esto ayudaría a los desarrolladores a comprender las responsabilidades y dependencias de cada componente.

DIAGRAMA DE COMPONENTES - Editor de archivo de texto, gestor de archivos y motor de plantillas son componentes. Permite ayudar a comprender las responsabilidades. Habla de dependencia y módulos, este diagrama sirve para documentar las relaciones entre el sistema a través de las interfaces.

6) Despliegue de una aplicación en la nube con microservicios:

Se requiere representar la infraestructura en la que se implementan los microservicios, incluyendo servidores, contenedores y bases de datos. Esto proporciona una visión clara de cómo los componentes del sistema se distribuyen y comunican entre sí.

DIAGRAMA DE DEPLOY + DIAGRAMA DE COMPONENTES - El primero suele ser utilizado junto al segundo, pues juntos dan una visión general de cómo estará desplegado el sistema de información. Es utilizado para ver cómo se sitúan estos componentes lógicos en los distintos nodos físicos, lo cual sería la infraestructura.

7) Implementación de un sistema de monitoreo y alerta:

Se requiere mostrar cómo los diferentes componentes del sistema interactúan para detectar, evaluar y notificar eventos de alerta. Esto permitiría a los desarrolladores comprender el flujo de información y la lógica de procesamiento de eventos en el sistema.

DIAGRAMA DE SECUENCIA - representar el intercambio de mensajes entre los distintos componentes. Este diagrama va de arriba a abajo, plasmando la lógica de como es el orden y proceso.

8) Diseño de un juego en línea multijugador:

Se requiere representar las entidades involucradas en el juego, como jugadores, personajes, objetos y enemigos. Además, un Diagrama de Secuencia podría ser utilizado para mostrar cómo las interacciones entre jugadores y objetos del juego ocurren en tiempo real. Estos diagramas ayudarían a los desarrolladores a entender la lógica del juego y a diseñar sus componentes de manera eficiente.

DIAGRAMA DE CLASE + DIAGRAMA DE SECUENCIA - Representar las entidades involucradas y diseñarlas con sus operaciones y tareas.

