

# UML

## 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.**

Se pueden visualizar las entidades clave como productos, clientes, carritos de compra y pedidos, junto con sus relaciones.

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 secuencia.**

En este caso, se puede ilustrar el flujo de información y eventos durante el inicio de sesión, registro y asignación de roles/permisos, lo que ayuda a los desarrolladores a entender cómo funcionan estas acciones en el sistema.

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.**

Debido a que representa los pasos y decisiones involucradas en el proceso.

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 facilitaría la comprensión de la arquitectura general y la gestión de dependencias.

### **Diagrama de paquetes.**

Ya que es útil para visualizar la arquitectura general de una aplicación y la gestión de dependencias entre diferentes módulos.

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.**

Debido a que se usa para describir la estructura y las interacciones entre los componentes de un sistema.

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 proporcionaría una visión clara de cómo los componentes del sistema se distribuyen y comunican entre sí.

**Diagrama de deploy.**

Para representar la infraestructura en la que se implementan los componentes del sistema, permitiendo visualizar la distribución de los componentes y cómo se comunican entre sí.

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.**

Ya que se quiere mostrar la interacción dinámica entre los componentes del sistema. Se puede ilustrar cómo los diferentes componentes interactúan para detectar, evaluar y notificar eventos de alerta.

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.**

Es adecuado para representar las entidades y sus relaciones en un sistema.