# **Formulario: Vista Lógica del Proyecto de Software**

**Proyecto:** LukasPet  
**Grupo:** 2 (Martin Godoy, Pablo Pereira, Francisco Zúñiga)  
**Fecha:** 18/10/2024

**1. Descripción General del Sistema**

Proporciona una descripción general del sistema y los principales componentes lógicos que lo forman.

**Descripción del sistema:**

El sistema es una plataforma de comercio electrónico que permite a los usuarios comprar productos para mascotas de manera rápida y eficiente. Está compuesto por varios módulos lógicos que manejan distintas funcionalidades, como la gestión de usuarios, el carrito de compras, el procesamiento de pagos, y la administración de inventario. Cada módulo está interconectado para ofrecer una experiencia fluida, desde el inicio de sesión del usuario hasta la confirmación de la compra.

**2. Principales Clases y Entidades del Sistema**

Describe las clases o entidades principales del sistema, sus atributos y métodos principales, y su propósito dentro del sistema.

**Clase 1 (Usuario):**Atributos: nombre, email, contraseña, rol, estado.   
Métodos: iniciar\_sesion(), cerrar\_sesion(), cambiar\_contraseñar(), actualizar\_perfil().  
Propósito: Gestiona la información de los usuarios del sistema y maneja la autenticación.

**Clase 2 (Producto):**Atributos: id\_producto, nombre, descripción, precio, stock, categoría.  
Métodos: agregar\_producto(), eliminar\_producto(), actualizar\_producto(), buscar\_producto().  
Propósito: Representa los productos disponibles en el sistema, permitiendo la gestión de su inventario.

**Clase 3 (CarritodeCompras):**Atributos: id\_carrito, lista\_productos, cantidad\_total, precio\_total.  
Métodos: agregar\_item(), eliminar\_item(), calcular\_total(), vaciar\_carrito().  
Propósito: Maneja la selección de productos por parte del usuario y facilita la gestión del proceso de compra.

**3. Relaciones entre clases**

Especifica cómo están relacionadas las clases entre sí, indicando relaciones de herencia, composición, agregación o dependencias.

**Relaciones entre clases:**

**Las clases principales del sistema están relacionadas de la siguiente manera:**

* **Usuario puede crear varios Pedidos.**
* **Un Pedido está compuesto por los productos seleccionados en el CarritoDeCompras.**
* **El CarritoDeCompras contiene varios Productos, los cuales se eliminan junto con el carrito.**
* **Cada Pedido está asociado a un Pago para completar la transacción.**

**4. Interfaces y dependencias**

Indica si alguna de las clases o entidades tiene interfaces y cómo interactúan con otros módulos del sistema a través de ellas.

**Descripción de interfaces y dependencias:**

Algunas clases del sistema implementan interfaces para asegurar la interoperabilidad entre los módulos. Las interfaces definen los métodos esenciales que deben ser implementados, permitiendo que diferentes componentes interactúen de manera consistente. Por ejemplo:

* Interfaz de Autenticación: Es implementada por las clases relacionadas con el módulo de inicio de sesión, facilitando la autenticación y validación de usuarios en otros módulos como el de gestión de usuarios y permisos.
* Interfaz de Gestión de Carrito: Permite que las clases del módulo de carrito de compras interactúen con el módulo de productos, asegurando la actualización de inventario y el cálculo de precios.

Las dependencias principales entre los módulos están gestionadas a través de estas interfaces, lo que garantiza una separación clara entre la lógica interna de cada componente y su interacción con el resto del sistema.

**5. Diagrama de Clases del Sistema**

Adjunta un diagrama de clases que representa la estructura lógica del sistema, las clases principales y las relaciones entre ellas.

**Diagrama de clases:**

**6. Diagrama de componentes (opcional)**

Si su sistema lo requiere, adjunte un diagrama de componentes que represente cómo los componentes principales del sistema están organizados y cómo interactúan.

**Diagrama de componentes:***(Adjunta el diagrama en esta sección)*

**7. Comentarios adicionales**

Incluye cualquier comentario adicional o aclaración relevante sobre la vista lógica de tu sistema.

**Comentarios:**

El diseño lógico del sistema prioriza la modularidad y la escalabilidad. Cada módulo ha sido desarrollado para ser fácilmente integrable con nuevas funcionalidades y adaptable a futuros cambios. Además, se ha prestado especial atención a la seguridad, especialmente en las áreas de autenticación y manejo de datos personales. Las entidades están optimizadas para mantener una interacción fluida y evitar redundancias, lo que contribuye a un sistema eficiente y robusto.