

# SISTEMA DE GESTIÓN DE UNA TIENDA EN LÍNEA

## Grupo 2

Gerson Rubén Quiroa del Cid – 2020000166

Mario Cesar Moran Porras – 202010793

Esdras Rodolfo Toc Hi – 201807373

Josué Daniel Míncuez Velásquez – 202003926



# REQUISITOS FUNCIONALES

1

## REGISTRO DE USUARIOS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam mattis, nunc vitae eleifend posuere, turpis mauris vestibulum purus, in pellentesque tellus elit vel nisl.

2

## CATÁLOGO DE PRODUCTOS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam mattis, nunc vitae eleifend posuere, turpis mauris vestibulum purus, in pellentesque tellus elit vel nisl.

3

## CARRITO DE COMPRAS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam mattis, nunc vitae eleifend posuere, turpis mauris vestibulum purus, in pellentesque tellus elit vel nisl.

4

## PROCESO DE COMPRAS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam mattis, nunc vitae eleifend posuere, turpis mauris vestibulum purus, in pellentesque tellus elit vel nisl.

5

## GESTIÓN DE INVENTARIO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam mattis, nunc vitae eleifend posuere, turpis mauris vestibulum purus, in pellentesque tellus elit vel nisl.

6

## ADMINISTRACIÓN DE PEDIDOS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam mattis, nunc vitae eleifend posuere, turpis mauris vestibulum purus, in pellentesque tellus elit vel nisl.

6 REQUISITOS GENERALES  
20 REQUISITOS ESPECÍFICOS

# REQUISITO NO FUNCIONALES:

## RENDIMIENTO

- *Tiempo de Respuesta*
- *Pruebas de estrés*
- *Caché y Almacenamiento en Memoria*
- *Balanceo de Carga*

## SEGURIDAD

- *Protección de Datos Personales*
- *Protección de Datos Financieros*
- *Autenticación Robusta*
- *Registro de Actividades y Auditoría*
- *Continuidad del Negocio*

## FIABILIDAD

- *Disponibilidad del Sistema*
- *Tolerancia a fallos*
- *Respuesta a Incidentes*

## DISPONIBILIDAD

- *Disponibilidad de 99.5% del tiempo mensual*
- *Mantenimiento en ventanas de baja demanda*
- *Plan de recuperación de desastres*

## MANTENIBILIDAD

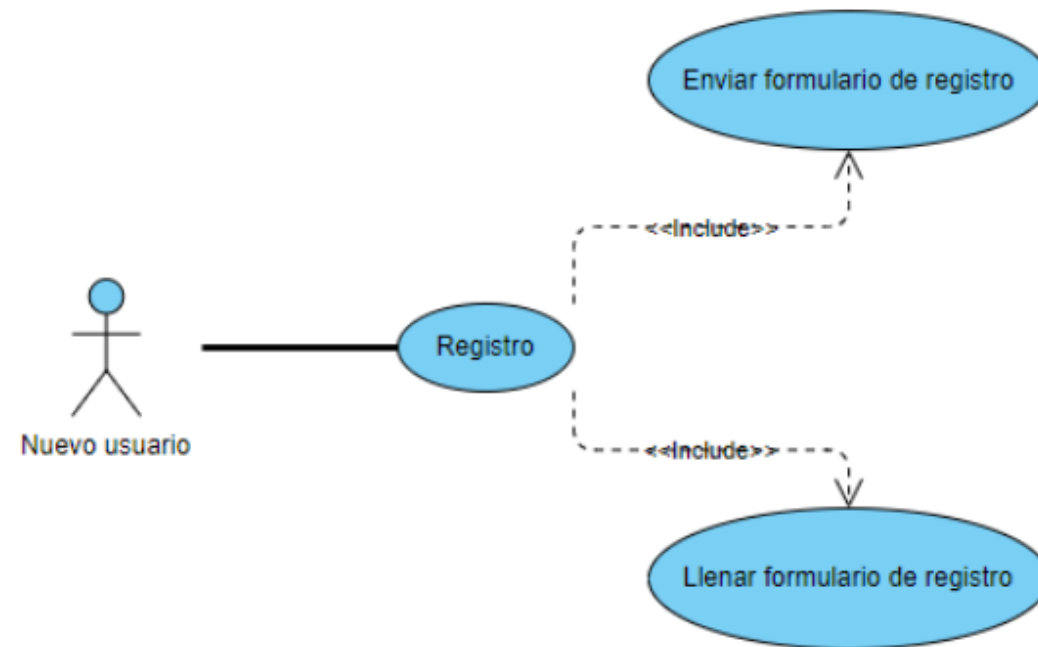
- *Identificación de tareas de mantenimiento necesarias*
- *Correctivo, adaptativo y perfectivo*
- *Frecuencia de mantenimientos*

## PORTABILIDAD

- *No más del 20% de sus componentes*
- *No más del 15% del código fuente total*
- *El sistema será desarrollado en PHP*

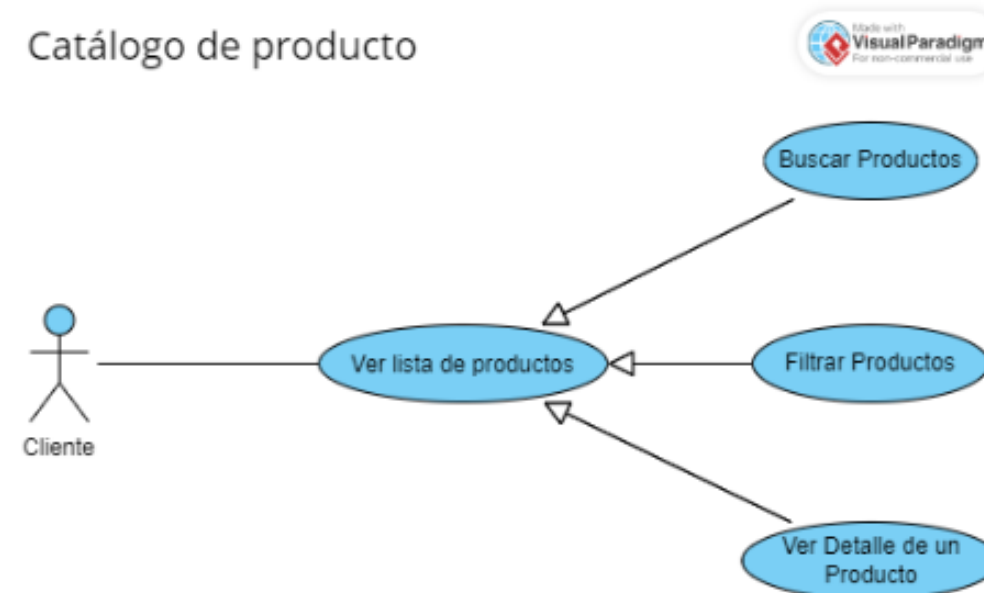
# CASOS DE USO

1. Registro de usuarios:

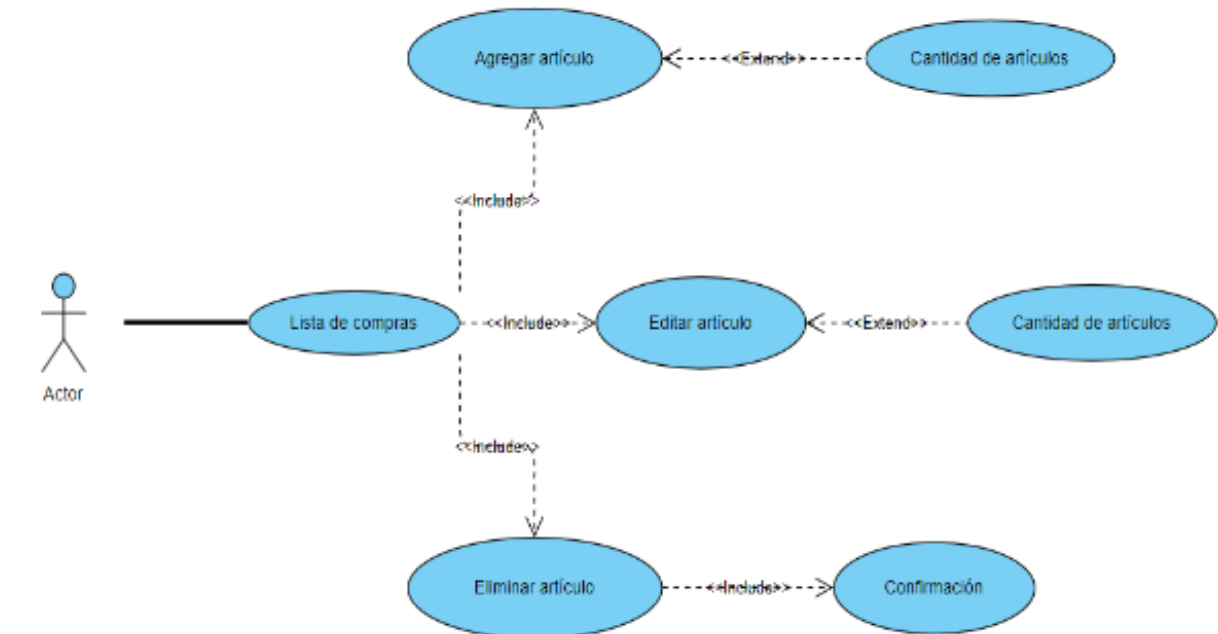


2. Catálogo de productos: Una vez que el usuario está registrado y ya ha ingresado a la tienda en línea el usuario puede ver el catálogo de productos.

## Catálogo de producto

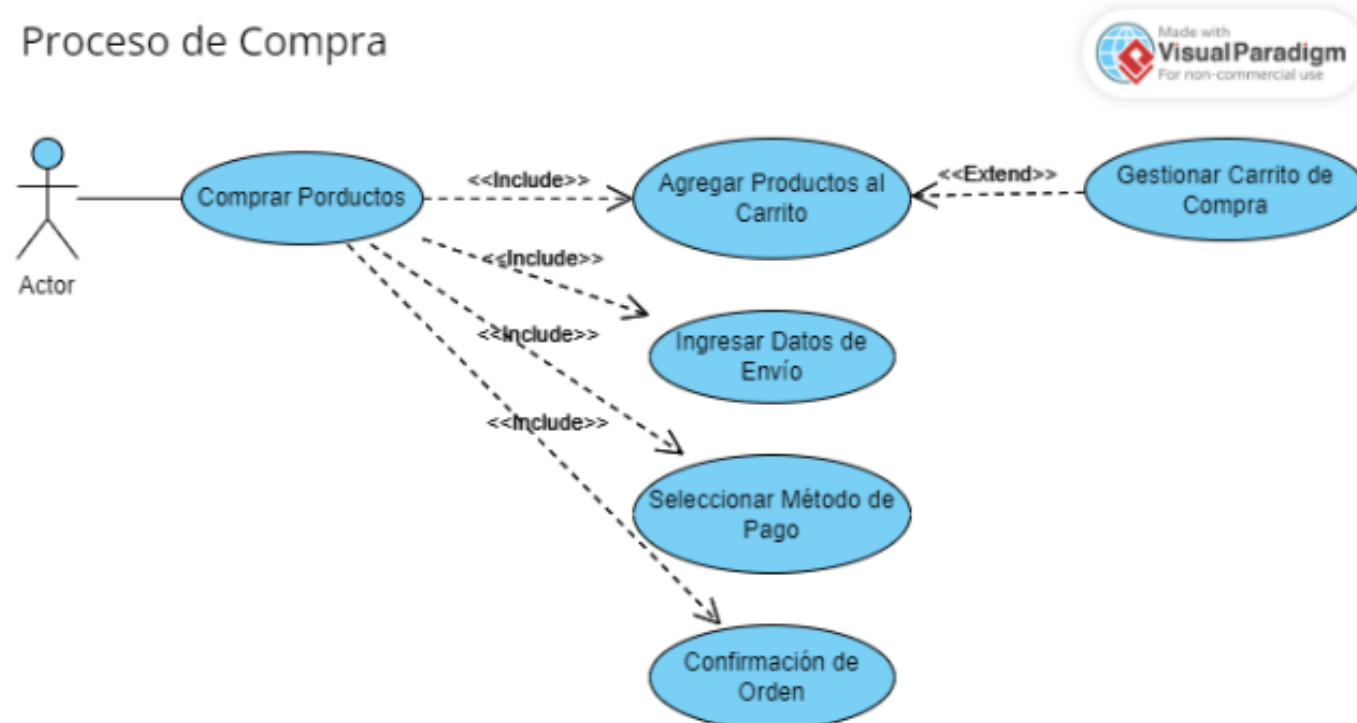


3. Carrito de compras:

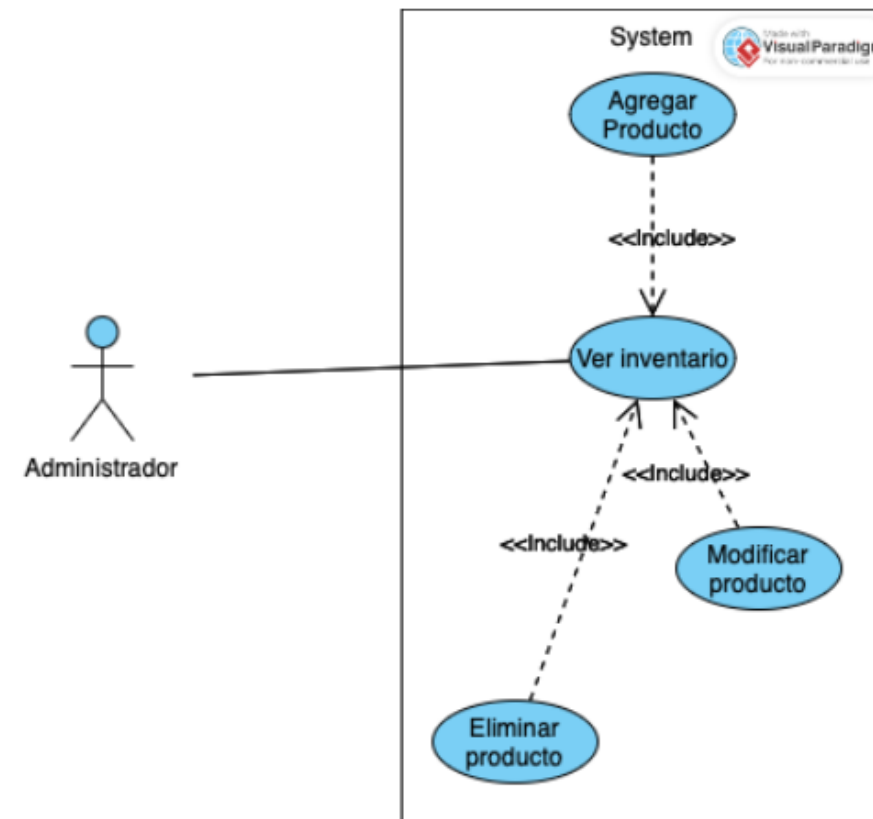


4. Proceso de compra: Una vez que el usuario ya eligió sus productos del catálogo y los agregó al carrito de compras, puede continuar con el proceso de compra.

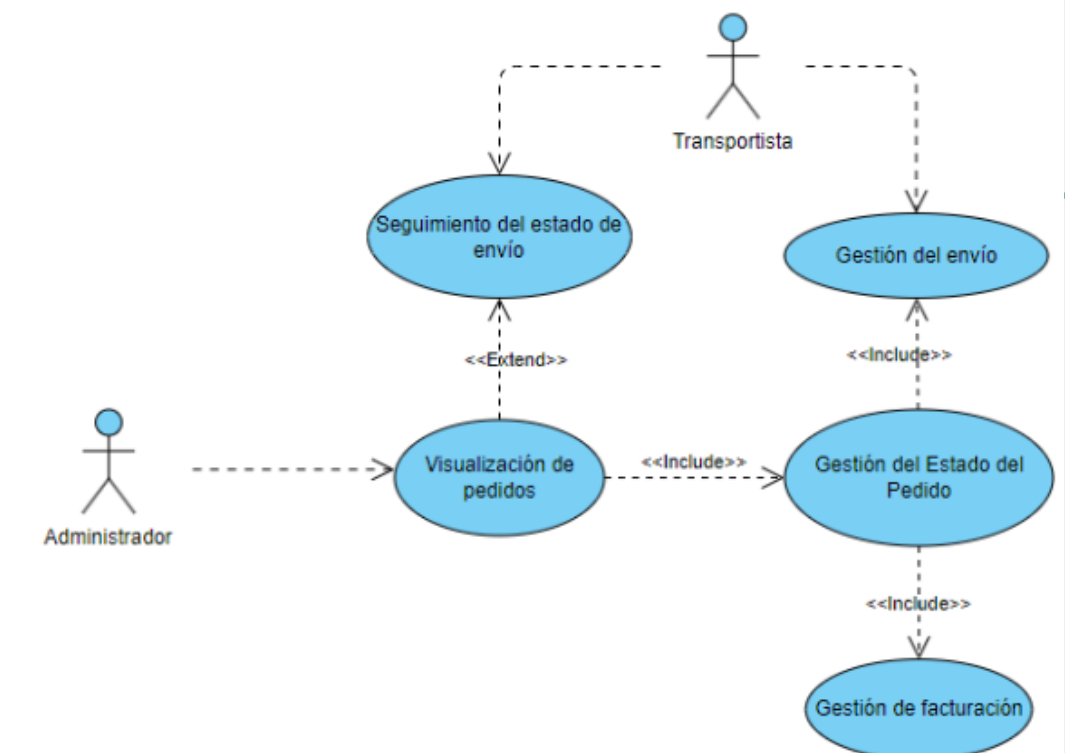
## Proceso de Compra



5. Gestión de inventario:



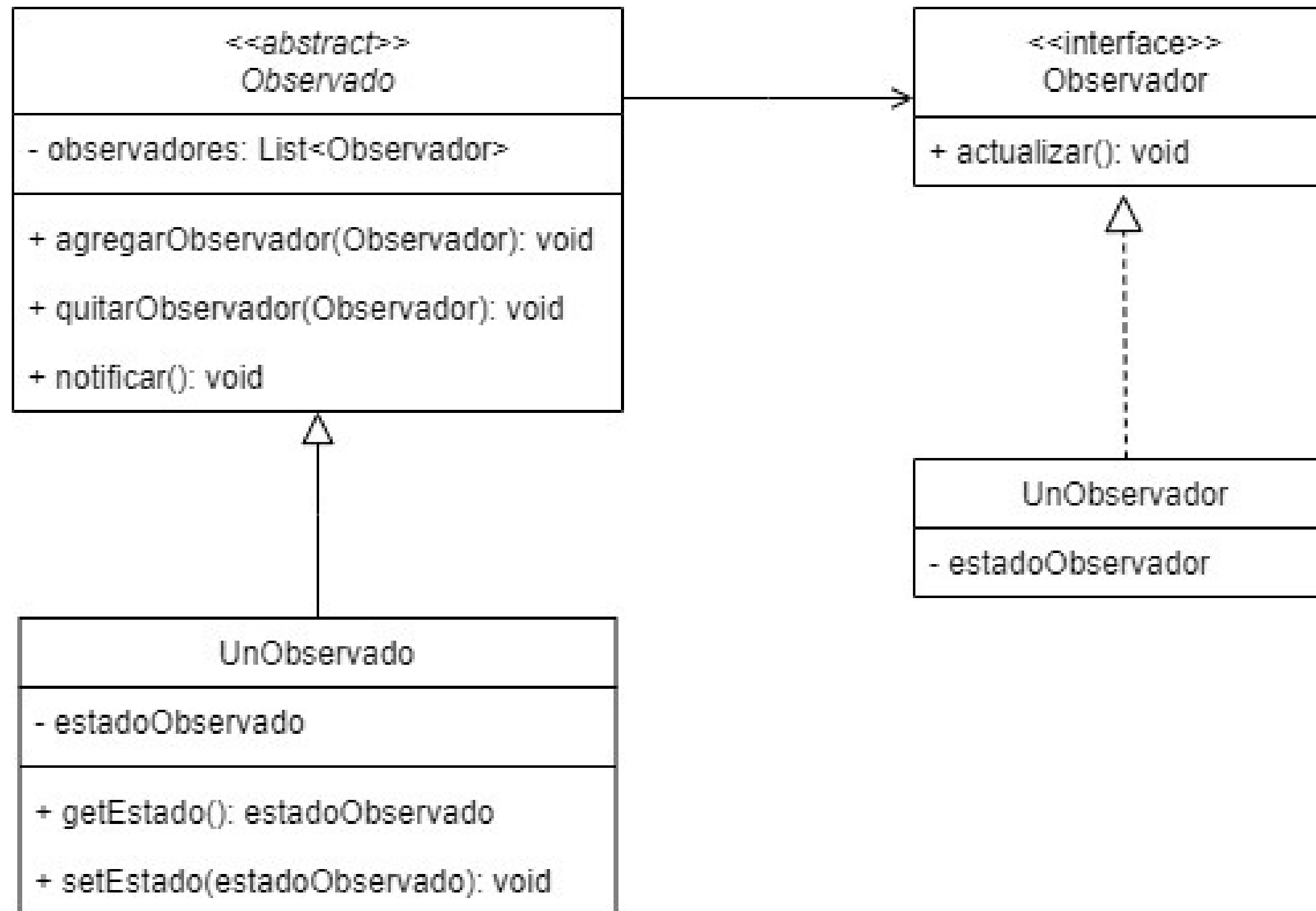
6. Administración de pedidos:



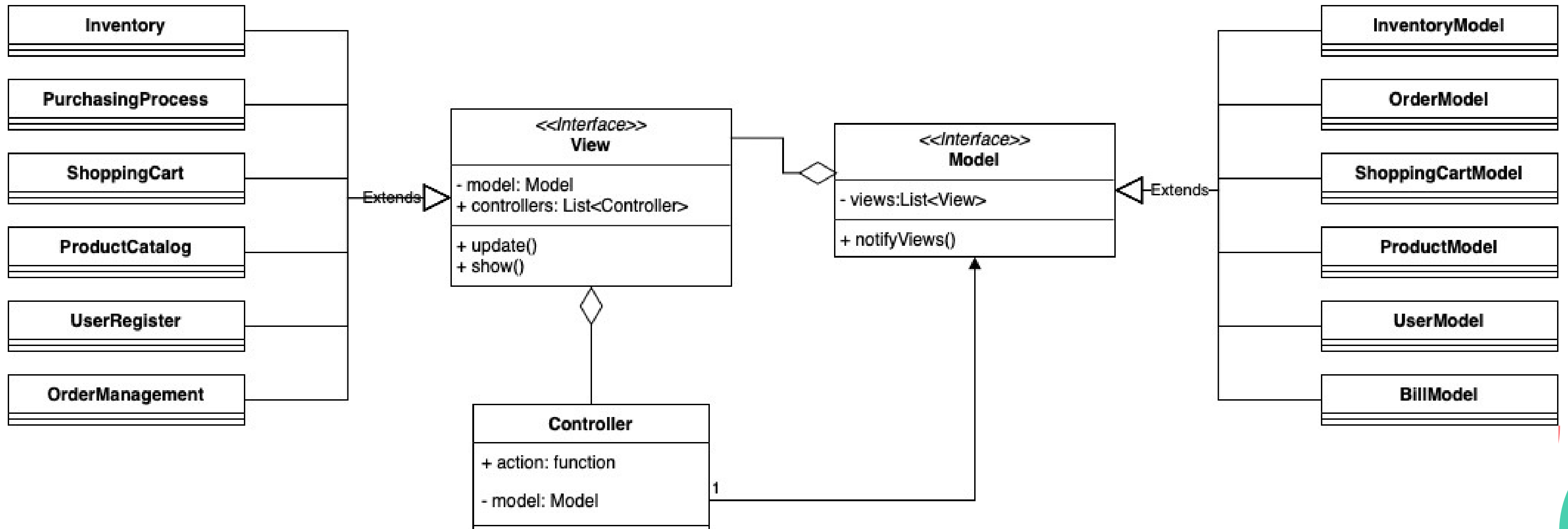
# CUADRO COMPARATIVO ELECCIÓN DEL FRAMEWORK DE PHP

Laravel	Symfony
<b><u>Curva de Aprendizaje:</u></b> Curva de aprendizaje de grado medio a fuerte.	<b><u>Curva de Aprendizaje:</u></b> Curva de aprendizaje un poco más severa.
<b><u>Flexibilidad:</u></b> Menos flexible en ciertos aspectos, pero proporciona muchas características listas para usar. Limitaciones de personalización. Tareas repetitivas automatizadas y migraciones de bases de datos simplificadas con Artisan	<b><u>Flexibilidad:</u></b> Altamente flexible y modular, permite la personalización extensa.
<b><u>Rendimiento:</u></b> Buen rendimiento, adecuado para aplicaciones de tamaño mediano a grande.	<b><u>Rendimiento:</u></b> Alto rendimiento, se adapta bien a aplicaciones de gran escala. Estable y robusto para aplicaciones empresariales a gran escala.
<b><u>Arquitectura:</u></b> Utiliza el patrón de arquitectura MVC.	<b><u>Arquitectura:</u></b> Basado en componentes, puede adaptarse a diferentes patrones de diseño (MVC, HMVC, etc.)
<b><u>Seguridad:</u></b> Ofrece características de seguridad integradas, pero requiere configuración adicional para algunas medidas específicas.	<b><u>Seguridad:</u></b> Enfoque en la seguridad, proporciona herramientas avanzadas para la protección de aplicaciones.
<b><u>Mantenimiento:</u></b> Facilidad para mantener y actualizar aplicaciones, gracias a su estructura clara.	<b><u>Mantenimiento:</u></b> Requiere un poco más de esfuerzo para mantener aplicaciones debido a su modularidad.
<b><u>Escalabilidad:</u></b> Escalable y permite el desarrollo de aplicaciones con crecimiento futuro.	<b><u>Escalabilidad:</u></b> Altamente escalable, adecuado para proyectos que necesitan crecer rápidamente.

# PATRON DE DISEÑO: OBSERVER



# DIAGRAMA DE CLASES



# PLAN DE PRUEBAS UNITARIAS

Número del Caso de Prueba	Componente	Descripción de lo que se Probará	Prerrequisitos
CA001	Módulo de registro	Verificar el registro exitoso de un nuevo usuario proporcionando información personal.	-
CA002	Módulo de Catálogo	Verificar la correcta visualización del catálogo de productos.	Catálogo con productos disponibles.
CA003	Módulo de Carrito	Verificar la adición exitosa de productos al carrito de compras.	Productos disponibles en el catálogo.
CA004	Módulo de Compra	Verificar el flujo completo del proceso de compra hasta la confirmación de la orden.	Productos en el carrito de compras.
CA005	Módulo de Inventario	Verificar la adición exitosa de nuevos productos al inventario.	-
CA005	Módulo de Pedidos	Verificar la visualización de pedidos realizados por los clientes.	Pedidos realizados por clientes.

## EJEMPLO

CA001						
Nº	Descripción	Método	Datos Entrada	Salida Esperada	¿OK?	Observaciones
1	Registro Exitoso	registerUser()	Datos válidos de un nuevo usuario (nombre, dirección, datos de contacto).	Confirmación de registro exitoso.	Si	Ninguna
2	Datos Incompletos	registerUser()	Datos incompletos de un nuevo usuario.	Mensaje de error indicando que faltan datos obligatorios.	Si	Ninguna
3	Datos Inválidos	registerUser()	Datos inválidos de un nuevo usuario.	Mensaje de error indicando que los datos no son válidos.	Si	Ninguna
4	Usuario Existente	registerUser()	Datos de un usuario ya registrado.	Mensaje de error indicando que el usuario ya existe.	Si	Ninguna
5	Contraseña Segura	registerUser()	Contraseña segura según políticas de seguridad.	Confirmación de registro exitoso.	Si	Ninguna



# PLAN DE PRUEBAS DE ACEPTACIÓN

- Validar de manera integral y exhaustiva la funcionalidad del Sistema de Gestión de una Tienda en Línea.
- Se busca garantizar la calidad y confiabilidad del sistema
- Se dirige a las unidades organizativas involucradas en el desarrollo, validación y puesta en marcha

Registro de un usuario	CP_001
<b>Descripción:</b> Verificar el registro exitoso de un nuevo usuario proporcionando información personal como lo es correo, usuario, teléfono, nombres y contraseña.	
<b>Prerrequisitos</b> Ninguno.	
<b>Pasos:</b> <ol style="list-style-type: none"><li>1. Acceder a la página de registro.</li><li>2. Ingresar datos personales.</li><li>3. Dar clic en botón 'registrarse'.</li></ol>	
<b>Resultado esperado:</b> Se espera que el sistema permita ingresar la información requerida y genere una confirmación de registro exitoso.	
<b>Resultado obtenido:</b> El sistema debe registró al usuario correctamente.	

# PLAN DE INTEGRACIÓN

Número del Caso de Prueba	Componente	Descripción de lo que se Probará	Prerrequisitos
CA001	Frontend - Backend	Registro de usuario	Ninguno
CA002	Backend - Base de datos	Registro de usuario	Enviar la información desde el frontend
CA003	Frontend - Backend	Agregar producto al inventario	Usuario administrador
CA004	Backend - Base de datos	Agregar producto al inventario	Usuario administrador

## EJEMPLO

CA001					
Paso	Descripción de pasos a seguir	Datos Entrada	Salida Esperada	¿OK?	Observaciones
1	Entrar a la página de registro	-	-	OK	Ninguna
2	Llenar el formulario de registro	Nombre, dirección, datos de contacto	-	OK	Que el frontend valide los inputs con regex
3	Enviar la información	Nombre, dirección, datos de contacto	-	OK	Que el frontend envíe la información correctamente
4	Registro correcto	-	Usuario registrado correctamente	OK	Mostrar un mensaje que le indique al usuario el registro exitoso

# PLAN DE PRUEBAS FUNCIONALES

- Verificar la funcionalidad del Sistema de Gestión de una Tienda en Línea.
- Objetivo principal es asegurar el funcionamiento de un flujo correcto e intuitivo.
- Se dirige a las unidades organizativas involucradas en el desarrollo, validación y puesta en marcha

Gestión de inventario agregar productos	CP-1	
	¿Prueba de despliegue?	No
Descripción: Los administradores deben ser capaces de realizar todas las operaciones fundamentales de un inventario, agregar productos.		
Prerrequisitos Perfil administrador.		
Pasos: <ul style="list-style-type: none"><li>- Iniciar sesión en la plataforma.</li><li>- Seleccionar la opción de agregar producto.</li><li>- Llenar el formulario de nuevo producto.</li><li>- Registrar el nuevo producto.</li></ul>		
Resultado esperado: El producto se agregue correctamente y se visualice en el catálogo.		
Resultado obtenido: Producto agregado al catálogo correctamente.		



# **DESAFÍOS Y LECCIONES APRENDIDAS**



### Desafíos:

- Encontrar un patrón de diseño que se acoplara a nuestras necesidades.
- Elaborar un correcto plan de pruebas para cada uno de los tipos de pruebas y llevarlos a cabo con escenarios realistas y relevantes.
- Seleccionar las tecnologías correctas que acompañen a nuestra base de código fuente.
- Redactar una documentación detalla, cuidando cualquier fragmento del mismo y a la bases puntualizar bien cada funcionalidad en la aplicación.



### Lecciones aprendidas:

- El uso de patrones de diseño no solo facilita la programación del software, también mejora la calidad del mismo haciéndolo mas fácil de mantener y actualizar, ya que el código es ordenado y refactorizado correctamente.
- Priorizar pruebas de funcionalidades complejas y priorización de datos que puedan ocupar los múltiples escenarios que se puedan dar.
- Una correcta elección de las tecnologías, puede facilitar la escalabilidad, mantenimiento y evolución de la aplicación.
- La documentación influye de manera positiva para recordar como es la construcción del software documentado y poder incluir nuevos elementos al equipo de forma sencilla. La importancia de tener documentación de manera que se pueda respaldar cualquier uso de tiempo en el desarrollo.



# GRACIAS POR LA ATENCIÓN

GRUPO 2  
ANÁLISIS Y DISEÑO 2