**Capstone**

SIGLAS Y SECCIÓN: **005D**

NOMBRE DE LA SEDE: **Antonio Varas**

**Documentación Técnica (Modelo 4+1 y Arquitectura)**

**Integrantes:**

**Tushar Mirwani / 21.149.877-3**

**Diego Infantas / 21.532.547-4**

**Jean Pier Huansha / 26.841.686-2**

**Profesor:**

[CINDY BETZABE CONTADOR CISTERNA](https://campusvirtual.duoc.cl/ultra/courses/_640454_1/outline/roster/edit/_21279805_1)

04/10/2025

Contenido

[Modelo de Vistas de Arquitectura 4+1 2](#_Toc210500437)

[Vista de Escenarios 3](#_Toc210500438)

[ Casos de Uso 3](#_Toc210500439)

[Vista Lógica 3](#_Toc210500440)

[ Diagramas de Secuencia (5) 4](#_Toc210500441)

[ Diagramas de Comunicación (3) 4](#_Toc210500442)

[ Diagrama de Clases 5](#_Toc210500443)

[Vista de Despliegue 5](#_Toc210500444)

[ Diagrama de Componentes 5](#_Toc210500445)

[ Diagrama de Paquetes 6](#_Toc210500446)

[Vista de Procesos 7](#_Toc210500447)

[ Diagramas de Actividad (1) 7](#_Toc210500448)

[Vista Física 8](#_Toc210500449)

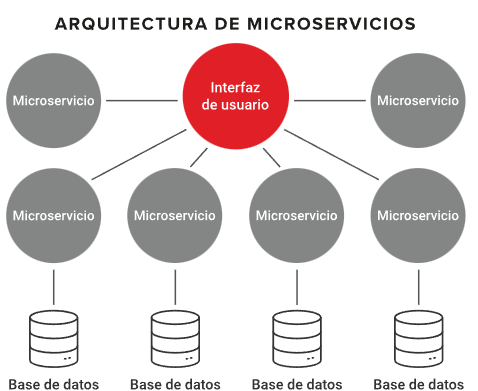
[ Diagrama de Despliegue 8](#_Toc210500450)

**Estilo Arquitectónico**

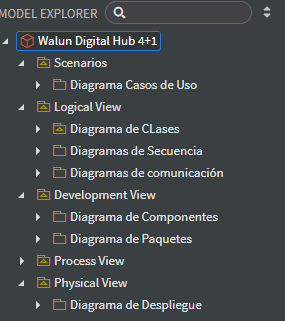
El estilo arquitectónico que usamos es la de **microservicios**, ya que permite dividir la aplicación en diferentes módulos independientes que se comunican entre sí mediante la **API**. Esta estructura se eligió porque nuestro sistema del e-commerce incluye varias funciones que pueden desarrollarse y mantenerse por separado, como la gestión de productos, pedidos, pagos, usuarios y despacho, facilitando la escalabilidad y la independencia de cada componente.

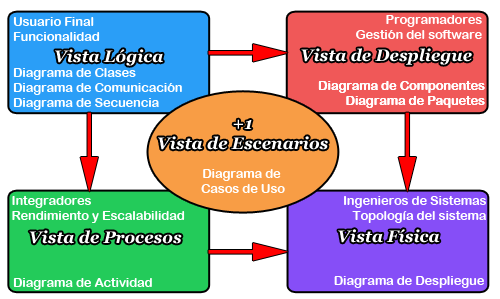
En nuestro proyecto, cada microservicio cumple una función específica: el de **productos** administra el catálogo y los precios, el de **pedidos** gestiona las compras y su estado, el de **pagos** se integra con **MercadoPago**, y el de **despacho** conecta con **Bluexpress** para generar y consultar el seguimiento de los envíos.

Este enfoque facilita la **actualización y ampliación del sistema**, haciendo posible agregar mejoras o nuevas funciones sin afectar los demás servicios, además de fortalecer la seguridad y la integración con futuras herramientas.



# Modelo de Vistas de Arquitectura 4+1

El modelo 4+1 organiza las vistas de nuestro sistema de sw para que distintos actores entiendan el proyecto desde sus intereses.



## Vista de Escenarios

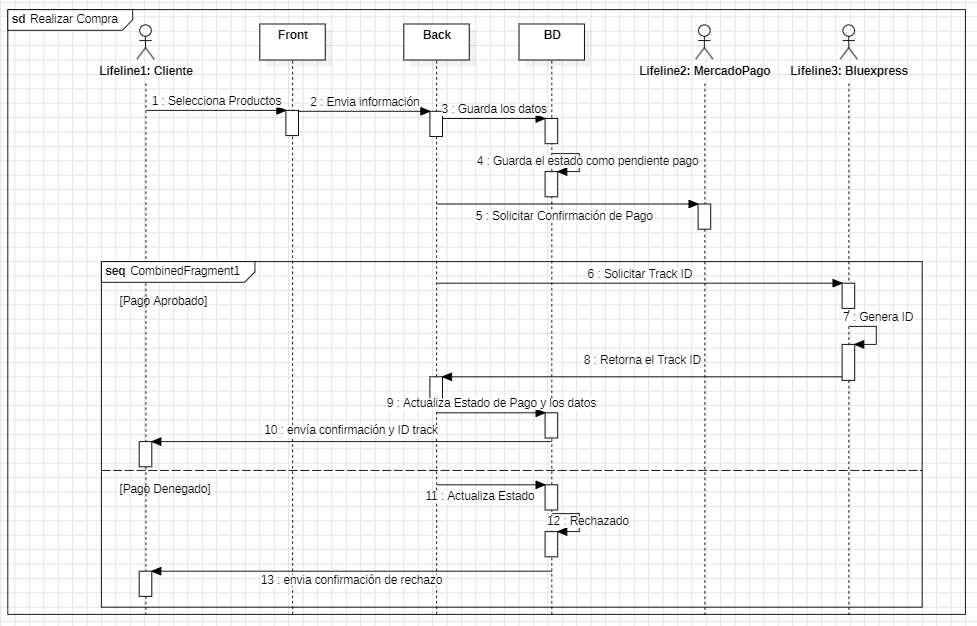
Mostramos cómo los usuarios interactúan con nuestro sistema y qué funcionalidades esperan, ya que con esto podemos capturar los requisitos funcionales y validar que la arquitectura **(microservicios)** que propusimos sea capaz de soportar los casos clave definidos para su funcionamiento.

### Casos de Uso

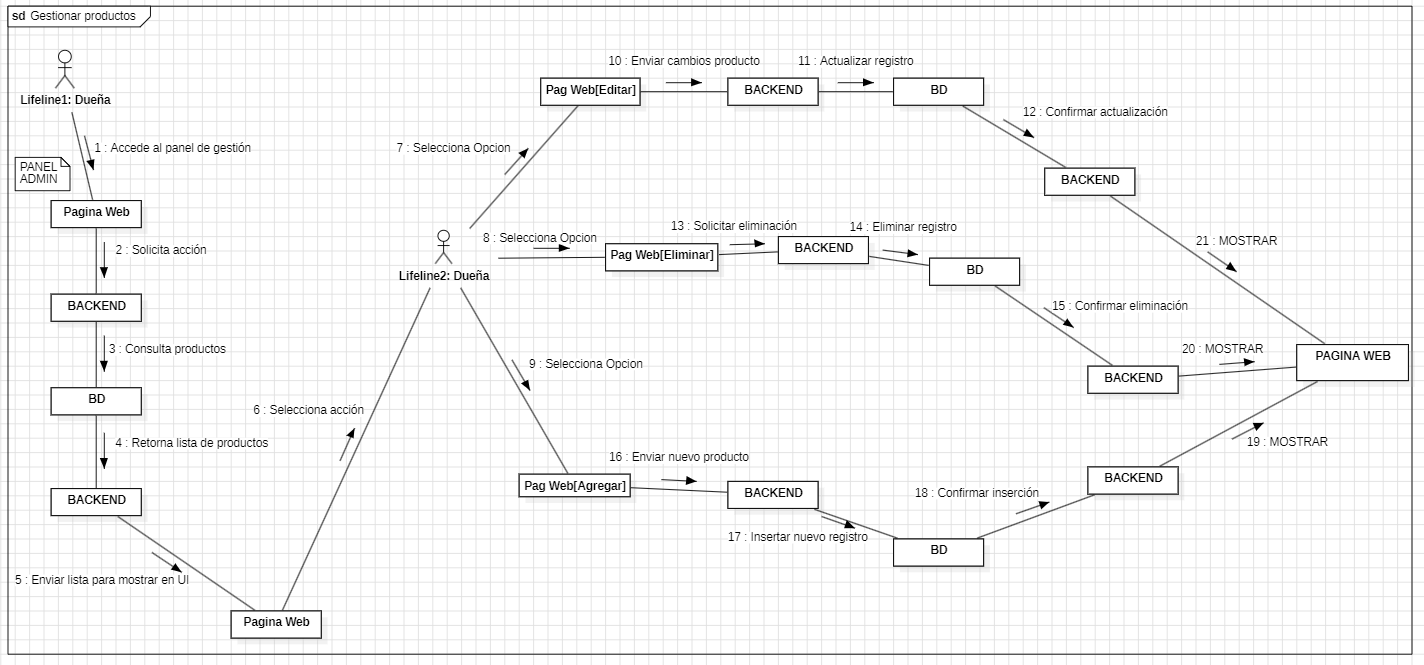
## Vista Lógica

En la vista lógica representamos la estructura estática de nuestro sistema, donde mostramos las **clases, entidades, relaciones y componentes principales**, ya que con esto podemos asegurar que los desarrolladores comprendan de manera clara la organización interna del proyecto.

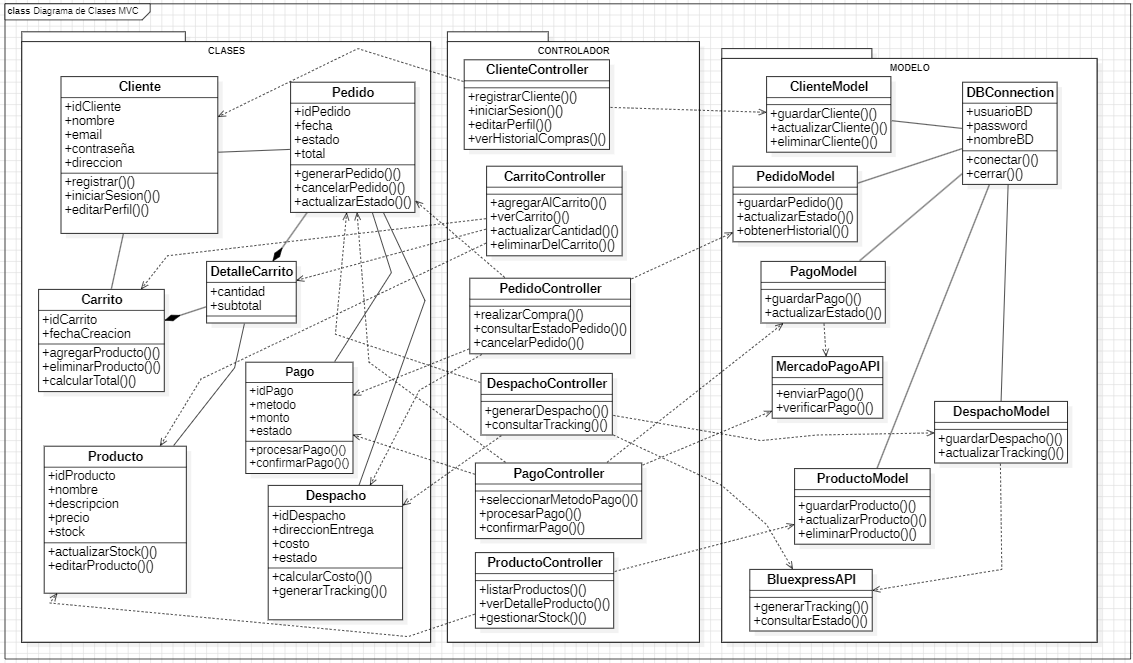
### Diagramas de Secuencia (5)

EJ: DG Sec. Realizar Compra 4+1

### Diagramas de Comunicación (3)

EJ: DG COM. Gestionar Productos 4+1

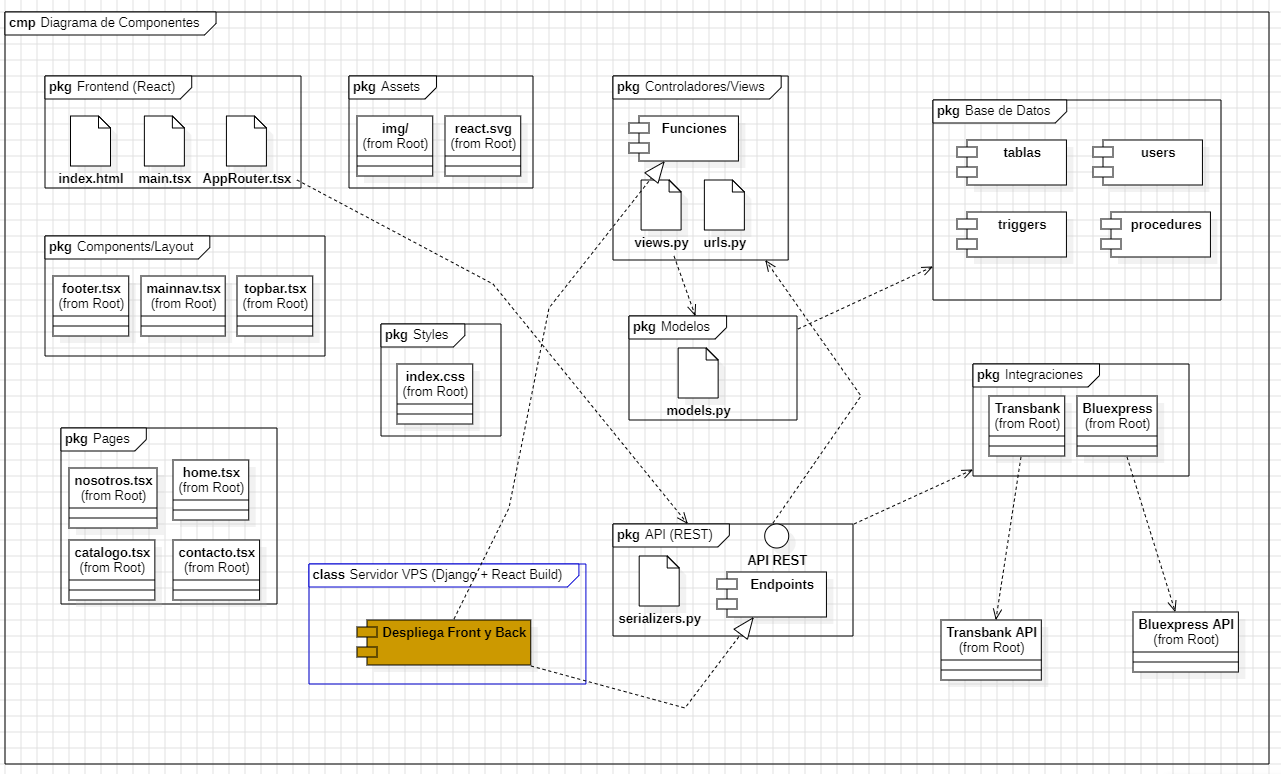
### Diagrama de Clases

EJ: **M**odelo **V**ista **C**ontrolador **(MVC)**

## Vista de Despliegue

En la vista de despliegue describimos cómo se desplegará físicamente nuestro sistema en el hardware y la red, lo que nos permite identificar servidores, conexiones, balanceadores, bases de datos y otros componentes de la infraestructura.

### Diagrama de Componentes



### 

### Diagrama de Paquetes

## Vista de Procesos

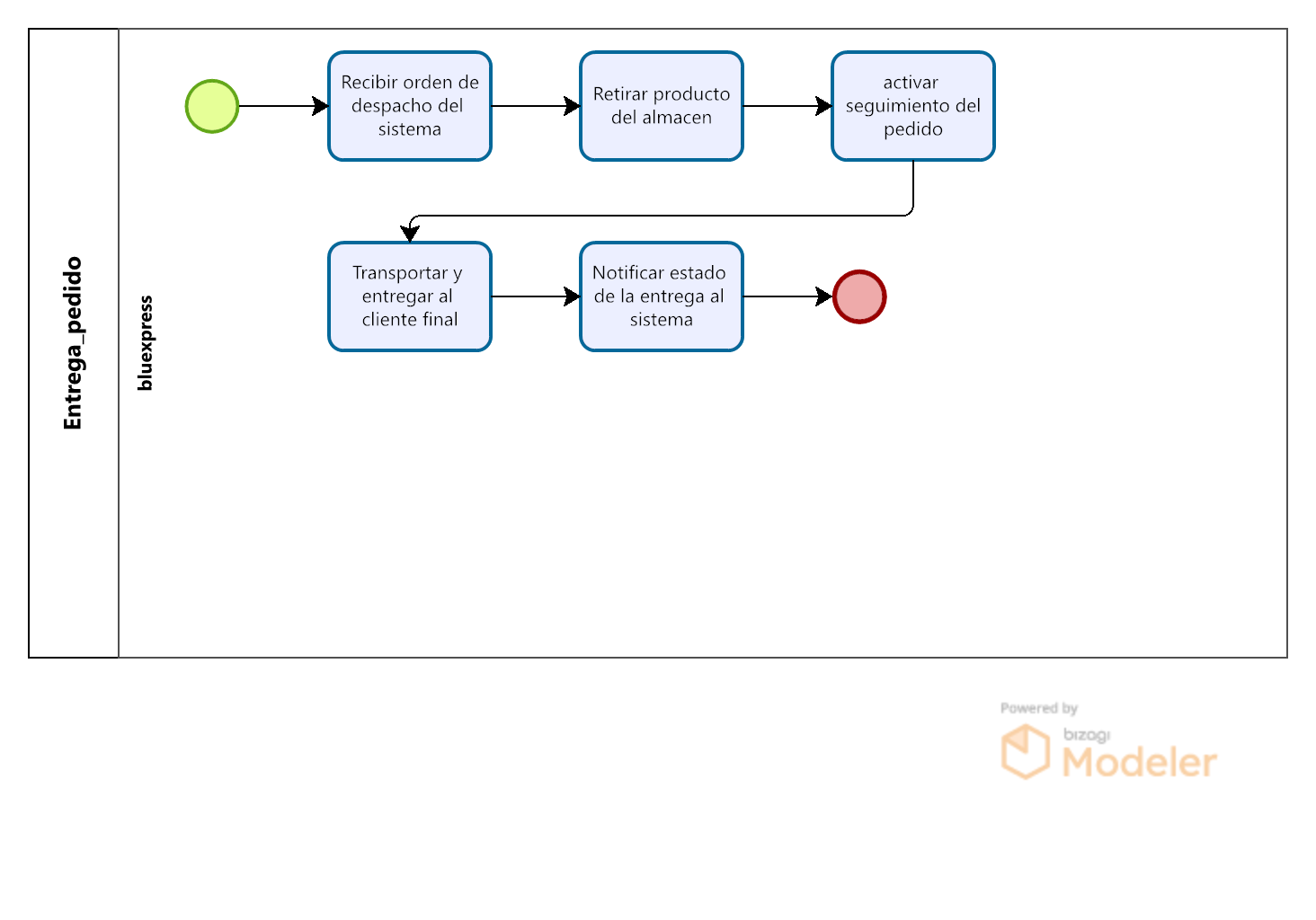
En la vista de procesos mostramos el comportamiento en ejecución de nuestro sistema, representando los procesos concurrentes, los hilos, la comunicación entre ellos y la sincronización, lo que resulta clave para analizar el rendimiento, la concurrencia y la escalabilidad de la solución que estamos desarrollando.

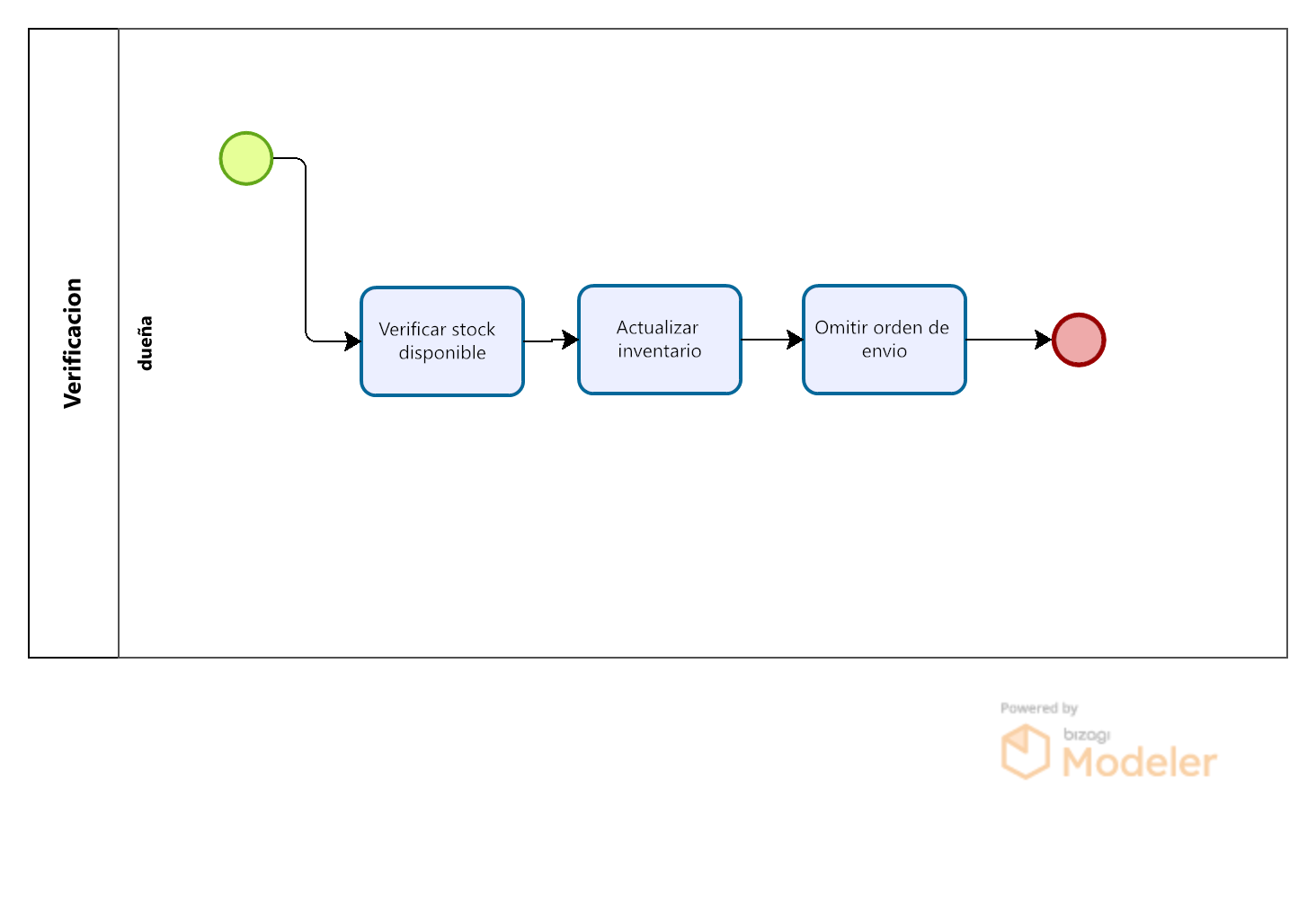
### Diagramas de Actividad (1)

EJ: Compra y Entrega de Productos [(ENLACE)](https://drive.google.com/file/d/1kDvI-U2qDMgn49sZ2ROj4cmYHgCc6Xgn/view?usp=sharing)

#### Flujo Principal (1)

#### Sub Procesos (4)





## Vista Física

En la vista física nos enfocamos en cómo se organiza el código fuente, las librerías, los módulos y los repositorios, ya que esto es fundamental para la gestión de la configuración y el control de versiones dentro de nuestro proyecto.

### Diagrama de Despliegue