

Asignatura	Datos del alumno	Fecha
<b>Metodologías, Desarrollo y Calidad en la Ingeniería de Software</b>	<b>Apellidos:</b> Pano Hernández	5 dic. 2022
	<b>Nombre:</b> Carlos	



### **Laboratorio**

# Extracción de las clases de análisis a partir de un documento de información con un enfoque orientado a objetos

### **Asignatura**

Metodologías, Desarrollo y Calidad en la Ingeniería de Software

### **Profesor**

Omar Uriel Domínguez Mendoza

### **Nombre del Alumno:**

Ing. Carlos Pano Hernández

### **Fecha de entrega:**

Lunes 5 de diciembre 2022

Asignatura	Datos del alumno	Fecha
<b>Metodologías, Desarrollo y Calidad en la Ingeniería de Software</b>	<b>Apellidos:</b> Pano Hernández	5 dic. 2022
	<b>Nombre:</b> Carlos	

### **Planteamiento del problema:** (continuación)

El equipo de *Azteca Sportswear* desea expandir su E-Commerce: *azteca-sportswear.com* al siguiente nivel:

Se desea crear una tienda que inicialmente contará con un máximo de 10,000 productos y estará conformada por varios vendedores locales. POR AHORA, solo se podrán vender camisetas, sudaderas y chamarras, sin embargo, se espera que en un futuro se agreguen muchas más categorías.

### **Desarrollo de modelos**

Continuando con el proceso de planeación de nuestro Software para *Azteca Sportswear* y tomando en cuenta que estamos trabajando bajo un esquema evolutivo, se procedió al desarrollo de las siguientes diagramas y modelos para aclarar algunas dudas de nuestros clientes.



Asignatura	Datos del alumno	Fecha
<b>Metodologías, Desarrollo y Calidad en la Ingeniería de Software</b>	<b>Apellidos:</b> Pano Hernández	5 dic. 2022
	<b>Nombre:</b> Carlos	

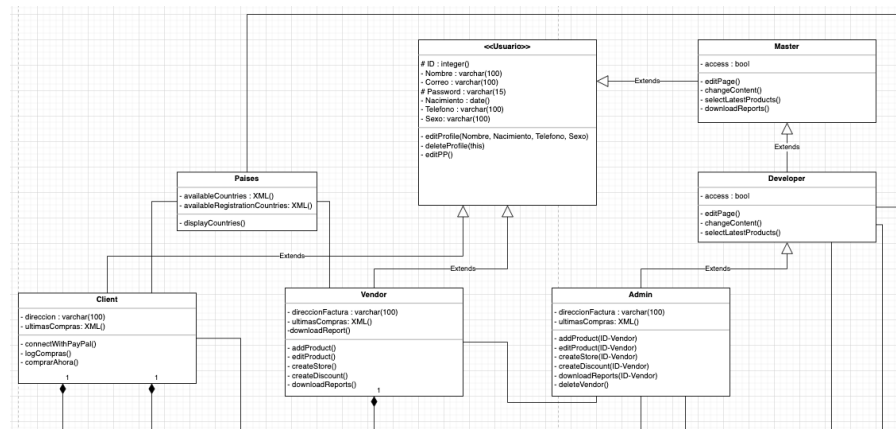


Diagrama 1.1. Acercamiento clase abstracta: “Usuario”

Podemos ver que de la clase “usuarios” derivan los clientes, vendedores y el Máster, un usuario padre que podrá tener acceso total. Cada uno con sus distintos comportamientos. Del Máster, derivan nuestros desarrolladores y administradores pues queremos que tengan las mismas funciones, pero serán restringidas.

Así pues, los clientes podrán hacer edición de sus productos, registrar su cuenta de PayPal y agregar productos a nuestro carrito de compras. Se creó la clase “países” para poder delimitar zonas de compra o bien, envíos.

Si seguimos con la siguiente parte del diagrama:

Asignatura	Datos del alumno	Fecha
<b>Metodologías, Desarrollo y Calidad en la Ingeniería de Software</b>	<b>Apellidos:</b> Pano Hernández	5 dic. 2022
	<b>Nombre:</b> Carlos	

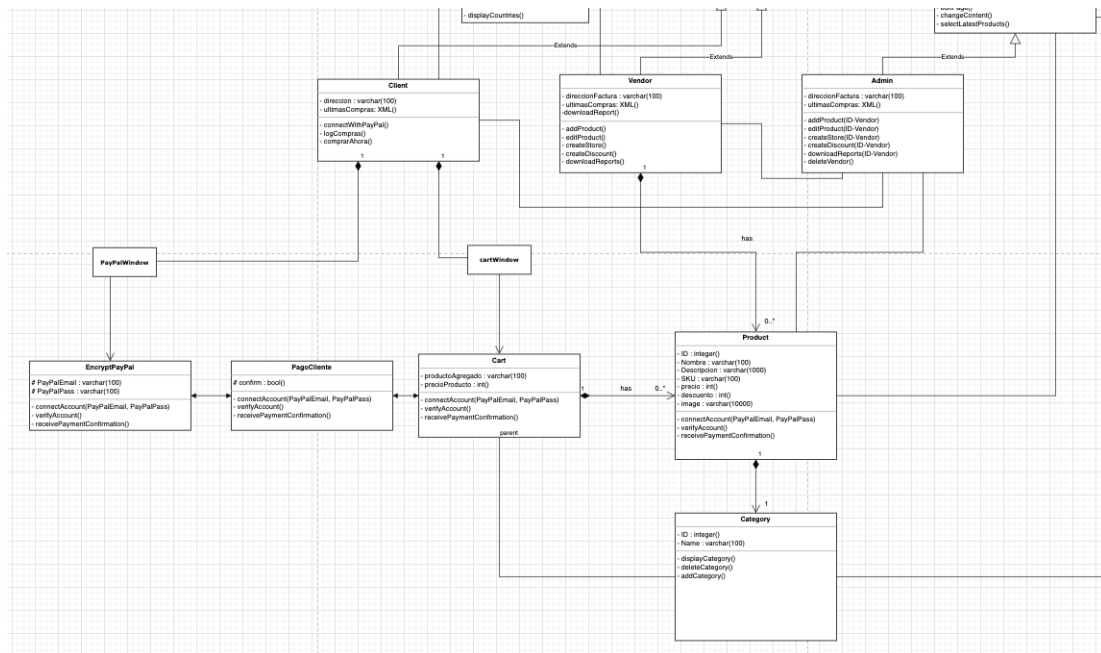


Diagrama 1.2. Acercamiento clase proceso de pago y carrito

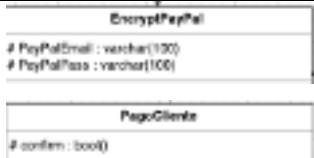
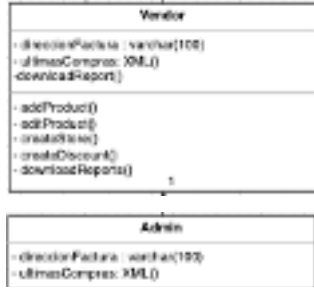
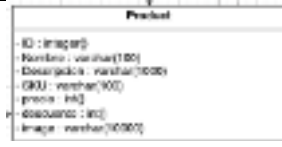
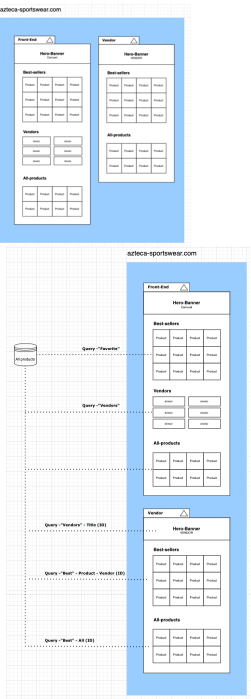
Vemos que un vendedor puede tener varios productos, pero si este usuario es eliminado, los productos no pueden existir. Continuando, el carrito de nuestro cliente puede contener varios productos de nuestro vendedor, y es así como el cliente puede comprar varios productos de múltiples “salers”.

Con esta recapitulación a nuestro diagrama de clases, podemos profundizar con el desarrollo del proyecto.

## Modelo MOF

Metamodelo	
MOF	
<b>M3: Capa del metamodelo</b>	<pre> classDiagram     class &lt;pre&gt;&lt;&lt;Usuarios&gt;&gt;     {         ID : integer()         Nombre : varchar(100)         Correo : varchar(100)         Password : varchar(15)         Nacimiento : date()         Telefono : varchar(100)         Sexo : varchar(100)     }     </pre> <p>Clase abstracta de usuarios</p>

Asignatura	Datos del alumno	Fecha
<b>Metodologías, Desarrollo y Calidad en la Ingeniería de Software</b>	<b>Apellidos:</b> Pano Hernández	5 dic. 2022
	<b>Nombre:</b> Carlos	

<b>M2: Capa del metamodelo</b>	
<b>M1: Capa del Modelo</b>	
<b>Mo: Capa de información</b>	 <p>Colección de datos con base a las preferencias de nuestro usuario en el producto seleccionado.</p>
<b>Interfaz</b>	

Asignatura	Datos del alumno	Fecha
<b>Metodologías, Desarrollo y Calidad en la Ingeniería de Software</b>	<b>Apellidos:</b> Pano Hernández	5 dic. 2022
	<b>Nombre:</b> Carlos	

A partir de nuestro diagrama de clases, podemos abstraer las distintas clases en nuestro metamodelo MOF.

Vemos que la interfaz otorgará a nuestro usuario las opciones necesarias para interactuar con nuestro sistema al desplegar la información de la capa M1. Posterior a ello, vemos como el proceso de compra se va rompiendo en las distintas capas y las mismas copas convergen entre si.

### Diagrama de Flujo

Con tal de tener un mayor entendimiento de la TOMA decisiones de nuestro Software, se realiza un diagrama de Flujo. Este se compone de secciones para visualizar la toma de decisiones y así facilitar la vista de casos o predecir comportamientos futuros de nuestros clientes:

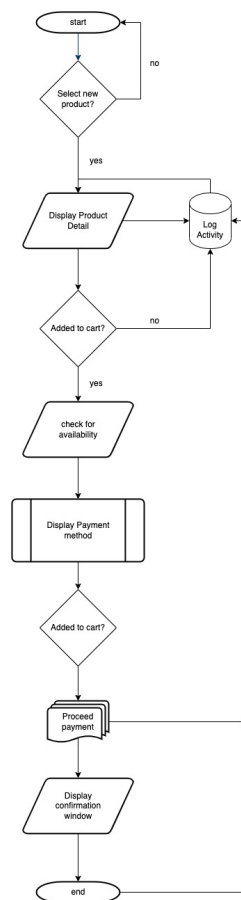


Diagrama de Flujo

Asignatura	Datos del alumno	Fecha
<b>Metodologías, Desarrollo y Calidad en la Ingeniería de Software</b>	<b>Apellidos:</b> Pano Hernández	5 dic. 2022
	<b>Nombre:</b> Carlos	

El diagrama anterior muestra el proceso perfecto que nuestro usuario idealmente debería seguir. Cabe recalcar que el mismo proceso puede generar frustración, ya que, si el usuario da clic varias veces a un mismo elemento, este puede verse afectado.

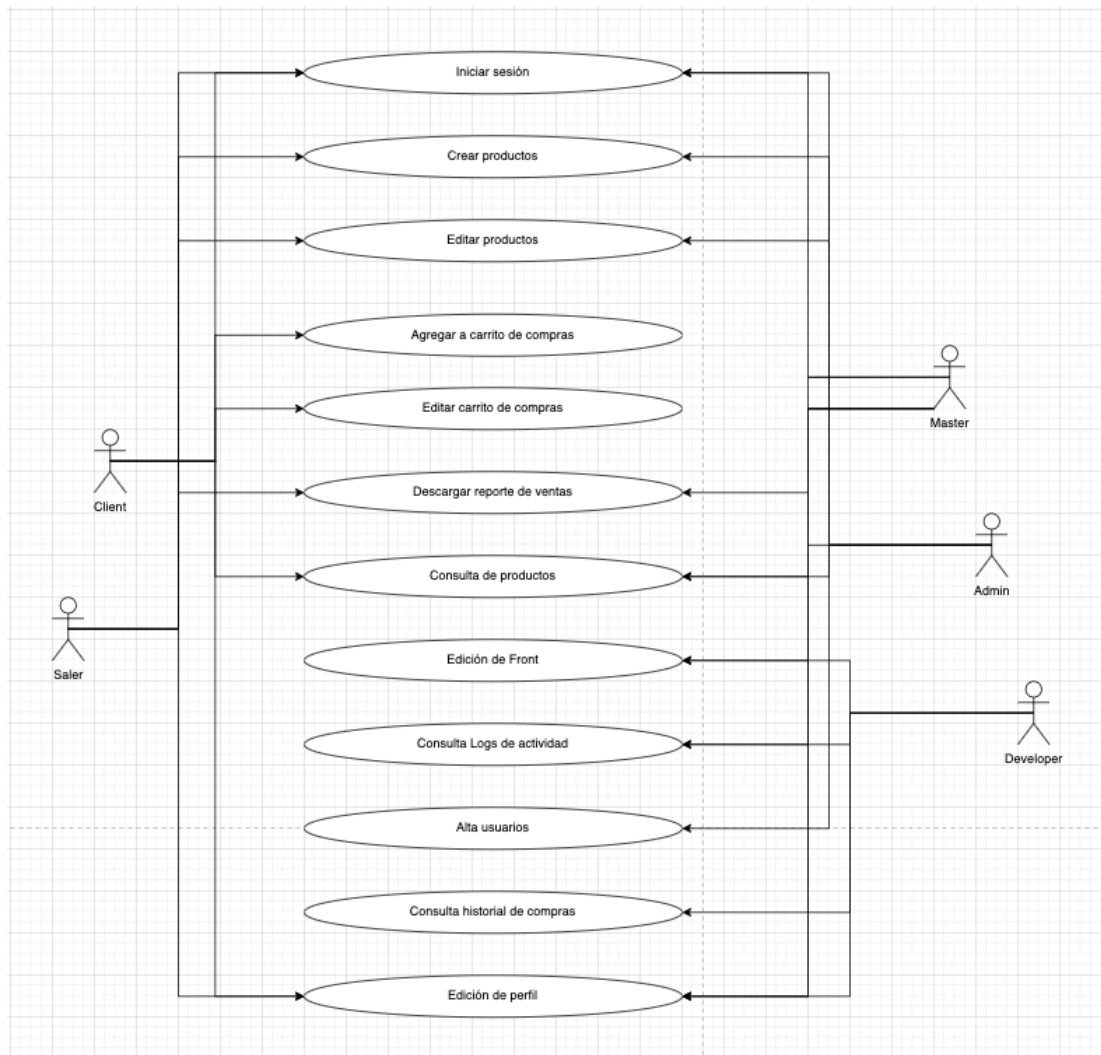
La calidad del sistema dependerá de la cantidad de Feedback que se otorgue a los desarrolladores. En el diagrama anterior, vemos un proceso simple, donde la toma de decisiones responde al estado actual de nuestros clientes.

### **Casos de uso**

(retomando nuestro trabajo anterior) A partir de nuestro sistema podemos obtener el siguiente Modelado de casos de uso:



Asignatura	Datos del alumno	Fecha
<b>Metodologías, Desarrollo y Calidad en la Ingeniería de Software</b>	<b>Apellidos:</b> Pano Hernández	5 dic. 2022
	<b>Nombre:</b> Carlos	



Modelado Casos de Uso

En el modelado anterior podemos observar las distintas actividades que nuestro usuario puede desempeñar en nuestro sistema. Dichos usos están dictados por los distintos requerimientos de nuestro “cliente” al principio del proyecto. Mismos que se enlistan a continuación:

- Una cuenta Master que permitirá administrar toda la tienda sin restricciones, así como agregar y eliminar usuarios.
- Un perfil de desarrolladores para mantener el código, sin necesidad de ver la información de nuestros clientes.
- Un perfil administrativo para poder dar soporte a clientes.

Asignatura	Datos del alumno	Fecha
<b>Metodologías, Desarrollo y Calidad en la Ingeniería de Software</b>	<b>Apellidos:</b> Pano Hernández	5 dic. 2022
	<b>Nombre:</b> Carlos	

- Nuestros clientes:
  - Podrán comprar productos de varios vendedores.
  - Podrán agregarlo a un carrito de compras.
  - Podrán hacer el pago correspondiente utilizando PayPal.
  - Así como la edición de todos sus datos.
- Nuestros vendedores:
  - Podrán agregar productos.
  - Editar o borrarlos.
  - Agregar descuentos.
  - Editar su información.

Con base en lo anterior, el sistema está pensado para ser escalable, por lo que los permisos podrán ser modificados cuando el cliente lo requiera.

Asignatura	Datos del alumno	Fecha
<b>Metodologías, Desarrollo y Calidad en la Ingeniería de Software</b>	<b>Apellidos:</b> Pano Hernández	5 dic. 2022
	<b>Nombre:</b> Carlos	

## Diagrama de secuencias

Entendiendo la dinámica de nuestros modelos, podemos generar un diagrama de secuencias para conceptualizar el timing de nuestro sistema, así como la sucesión de casos. Vemos en esta representación gráfico el comportamiento de nuestro software de manera vertical para facilitar la lectura de procesos concurrentes y asíncronos.

A continuación, mostramos el resultado:

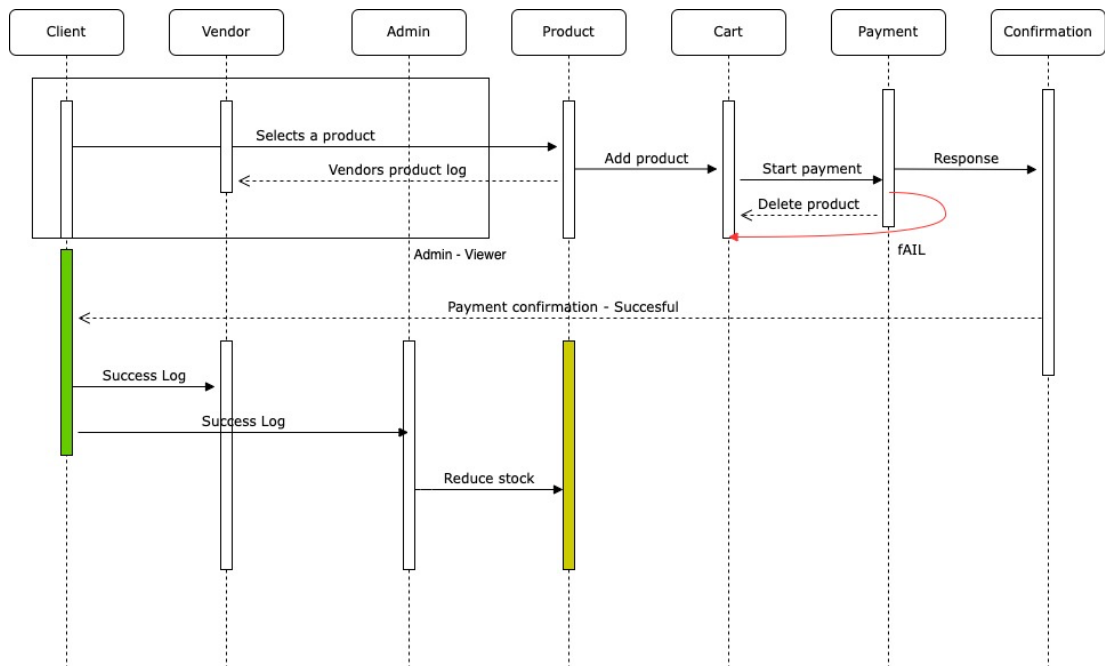


Diagrama de secuencias

Dentro de nuestro diagrama de secuencias, podemos notar el proceso de agregar un producto a nuestro carrito, la interacción y el papel de nuestro administrador como “viewer”.

Notamos que el proceso de pago no tiene tanta prioridad, puesto que es externo con PayPal y nosotros solo recibimos respuesta del API.

Asignatura	Datos del alumno	Fecha
<b>Metodologías, Desarrollo y Calidad en la Ingeniería de Software</b>	<b>Apellidos:</b> Pano Hernández	5 dic. 2022
	<b>Nombre:</b> Carlos	

Una vez teniendo la respuesta, podemos notificar a nuestro Cliente que la compra fue realizada con éxito y finalmente reducir el stock para no sobrevender nuestro producto.

### Diagrama de componentes

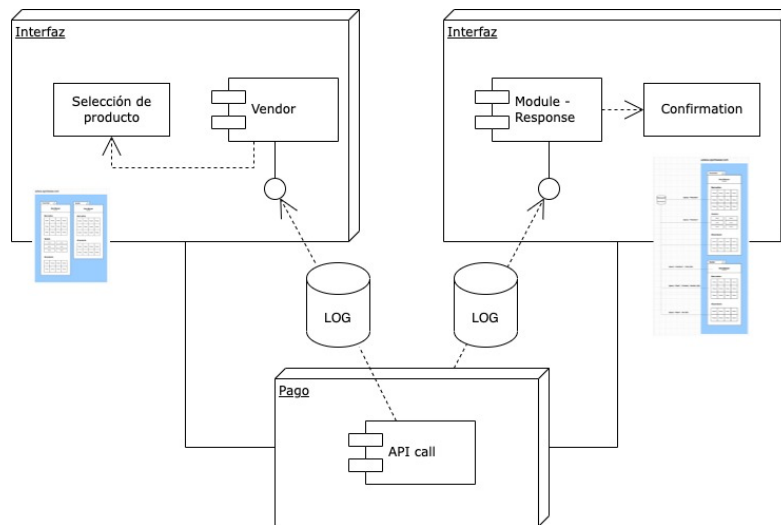


Diagrama de componentes

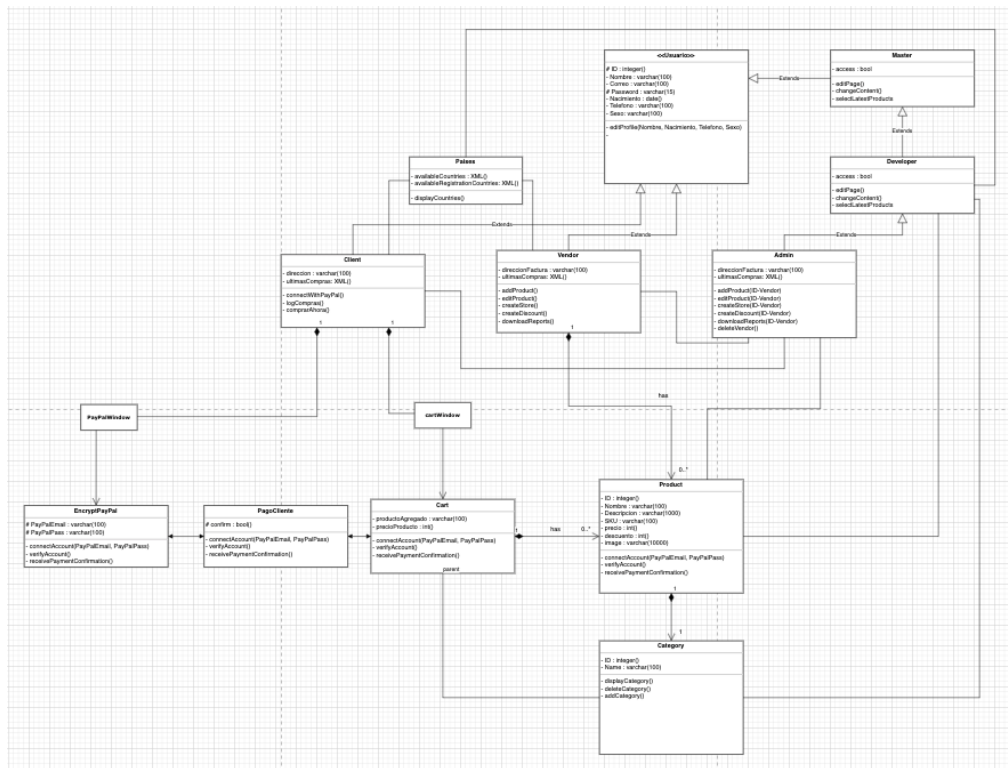
A partir de nuestro diagrama de secuencias, vemos que nuestro sistema está modularizado de manera general por estos componentes.

Una interfaz que provee de la visualización de todos los datos, el middleware de pago y la interfaz de confirmación. Dicha práctica de modularizar nuestro sistema en estos componentes abstractos nos hace ver igual que el registro de las actividades es constante, de hecho, esta vista simplifica a gran escala la complejidad del proyecto.

### Diagrama de entidad y relación

Tomando en cuenta que la base de datos se obtendrá de nuestro diagrama de clases. Las columnas estarán delimitadas por los atributos de cada uno de los diagramas:

Asignatura	Datos del alumno	Fecha
<b>Metodologías, Desarrollo y Calidad en la Ingeniería de Software</b>	<b>Apellidos:</b> Pano Hernández	5 dic. 2022
	<b>Nombre:</b> Carlos	



## Conclusión

A partir de este trabajo comprendí el uso y eficacia de otros diagramas no conocidos. Me gusta el desarrollo del proyecto, así como la continuación del trabajo. Para futuros casos, tal vez dividir la cantidad de diagramas en distintas tareas con tal de no hacer la misma tan larga.

El uso de representaciones gráficas, ayuda al cliente o Stakeholders a un mayor entendimiento del proyecto, lo cual facilita su lectura y posibles mejoras a futuro.

## Bibliografía

- Booch, G., Rumbaugh, J. y Jacobson, I. (2005). Unified Modeling Language User Guide (2° ed.). Addison Wesley.
- IEEE. (1998). IEEE 830-1998 - IEEE Recommended practice for software requirements specifications [Standard]. The Institute of Electrical and Electronics Engineers.<https://standards.ieee.org/standard/830-1998.html>

Asignatura	Datos del alumno	Fecha
<b>Metodologías, Desarrollo y Calidad en la Ingeniería de Software</b>	<b>Apellidos:</b> Pano Hernández	5 dic. 2022
	<b>Nombre:</b> Carlos	

- Jacobson, I., Booch, G. y Rumbaugh, J. (2000). El proceso unificado de desarrollo de software. Addison Wesley.