Social E-Commerce

Documento de Arquitectura de Software

Versión 1.0

Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| <27/Mayo/18> | 1.0 | Versión inicial | Grupo Dijkstra |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contenidos

1. Introducción 2

1.1 Definiciones, Siglas y Abreviaturas 2

1.2 Referencias 2

2. Representación Arquitectónica 2

3. Objetivos de la Arquitectura y Restricciones 2

4. Vista de Casos de Uso 2

4.1 Realizaciones de los Casos de Uso 2

5. Vista Lógica 2

5.1 Visión General 2

5.2 Paquetes de Diseño Significativos 2

6. Vista de Procesos 2

7. Vista de Despliegue 2

8. Vista de Implementación 2

8.1 Visión General 2

8.2 Capas 2

9. Vista de Datos (opcional) 2

10. Tamaños y Performance 2

11. Calidad 2

Documento de Arquitectura de Software

# Introducción

En vista a la construcción del sistema propuesto, este documento persigue especificar la estructura general de la solución a fin de tener un entendimiento general sobre las funcionalidades y características que serán brindadas.

Habiendo planteado el objetivo de este documento, procedemos a continuación a describir su alcance.

* Representación arquitectónica: Describe el estilo arquitectónico elegido para la construcción del sistema.
* Vista de casos de uso: Identifica las funcionalidades principales a entregar.
* Vista lógica: Presenta el sistema desde el punto de vista de las clases de diseño.
* Vista de procesos: Describe la estructura de procesos centrales, focalizando tanto en la estructura del proceso o los estados atravesados por un objeto de interés.
* Vista de despliegue: Presenta la forma en la que va a estar distribuido físicamente el sistema desarrollado, detallando los recursos de hardware que se requieren y la conectividad que debe existir entre ellos.
* Vista de implementación: Describe la estructura macro del sistema en la forma de componentes de alto nivel.
* Tamaños y performance: Establece las principales métricas de rendimiento y hardware recomendados para utilizar de forma eficaz el sistema.
* Calidad: Presenta los atributos de calidad que se derivan de la arquitectura delineada para el sistema.

## Definiciones, Siglas y Abreviaturas

Owner: Persona la cual ofrece un producto para su venta.

Círculo de compra: Conjunto de personas que aportan un monto determinado de dinero a un pozo, que en una fecha previamente fijada, se realizará un sorteo y el ganador se llevará todo el pozo.

Grupo de compra: Conjunto de personas con la intención de adquirir un producto de un Owner a un precio especial.

## Referencias

* Documento objetivo, límite y alcance.docx
* TP Integrador - Social Commerce

# Representación Arquitectónica

Al momento de empezar a definir la macro estructura del sistema, se hizo evidente que:

* El punto de acceso sería exclusivamente a través de tecnologías web
* El modo de operación estaría enfocado a un esquema cliente/servidor.
* La complejidad del sistema es media/alta, con lo cual se debe poder manejar.
* Es esencial asegurar la escalabilidad y mantenibilidad a largo plazo.

De acuerdo con estas observaciones, quedó establecido que el estilo arquitectónico que mejor se adapta a las necesidades del sistema es el de un modelo en capas. Profundizando sobre esta idea, este estilo será llevado a la práctica a través del patrón MVC, pues el mismo se adapta perfectamente a un entorno web y provee la flexibilidad necesaria para poder construir un sistema robusto y mantenible.

# Objetivos de la Arquitectura y Restricciones

**OBJETIVO**

**Se quiere un sistema el cual conecte vendedores (oferta) y compradores (demanda) de manera específica y puntual, brindando así una gran probabilidad de éxito de que se concrete la operación de venta y obtener ganancias de esta.**

**LÍMITE**

* El sistema no realiza logística de envió
* El sistema no permite pagos contrarrembolso
* El sistema no registra usuarios de redes sociales
* El sistema solo permite pago en pesos argentinos
* El dinero otorgado en círculos solo puede ser utilizado para las compras en la plataforma

**ALCANCE**

* **Módulo Catálogo**
* - Agregar producto
* - Eliminar producto
* - Buscar productos
* - Consultar producto
* - Eliminar producto
* - Modificar producto
* - Visualizar catalogo
* - Eliminar el catalogo

* **Modulo Carrito de Compras**
* - Agregar producto al carrito
* - Eliminar producto del carrito
* - Visualizar carrito
* - Comprar productos del carrito
* **Módulo de Envíos**
* - Visualizar envió
* - Calcular e informar costo y tiempo de envió

* **Modulo Círculos**
* - Unirse a circulo
* - Salir de un circulo
* - Aportar dinero
* - Realizar sorteo
* - Notificar ganador
* - Asignar premio

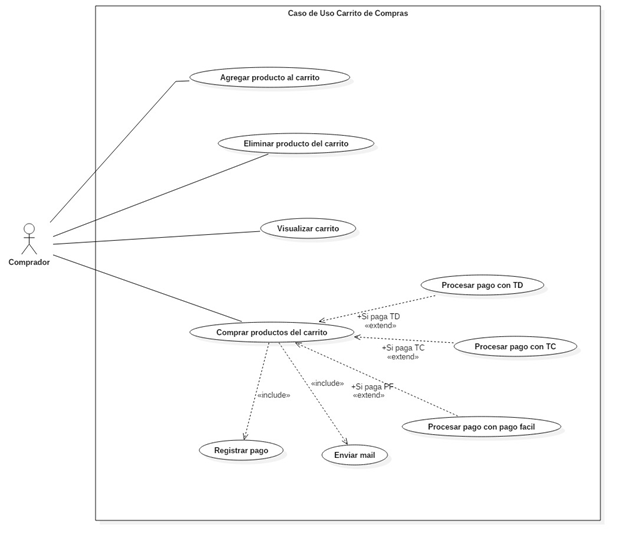
* **Modulo Equipos de compra**
* - Entrar en equipo de compra
* - Comprar con equipo
* - Ver historial de compras de equipo
* - Proponer equipo de compra
* - Salir de equipo de compra
* - Modificar acuerdo
* - Alertar equipo por compra
* **Modulo Usuarios. Compradores**
* - Iniciar sesión
* - Registrar comprador
* - Modificar comprador
* - Dar de baja comprador
* - Vincular cuenta a redes sociales
* - Visualizar datos comprador
* - Cerrar sesión
* **Módulo Usuarios. Vendedores**
* - Iniciar sesión
* - Registrar vendedor
* - Modificar vendedor
* - Dar de baja vendedor
* - Visualizar datos vendedor
* -Cerrar sesión
* - Visualizar publicaciones
* **Módulo Compra**
* - Realizar compra
* - Cancelar compra
* - Visualizar compras
* **Módulo Venta**
* - Realizar publicación
* - Cancelar venta
* - Consultar venta
* - Ver historial de ventas
* **Módulo Subastas**
* - Registrar subasta
* - Cancelar subasta
* - Notificar ganador subasta
* - Realizar oferta en subasta
* - Asignar producto
* - Finalizar subasta
* **Módulo Ofertas**
* -Realizar oferta
* -Visualizar ofertas activas
* - Dar de baja una oferta
* -Modificar oferta
* **Módulo Pagos**
* -Tarjeta de débito
* -Tarjeta de crédito
* - Pago fácil
* - Registrar pago

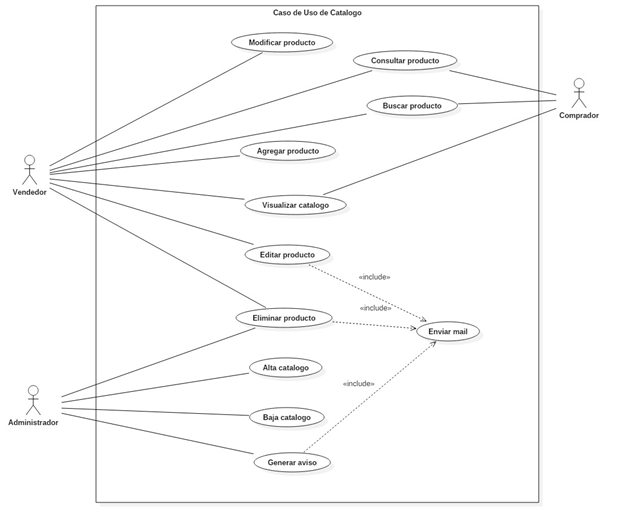
* **Módulo Reclamos**
* - Iniciar un reclamo
* - Visualizar estado de reclamo
* - Cancelar reclamo
* - Finalizar reclamo

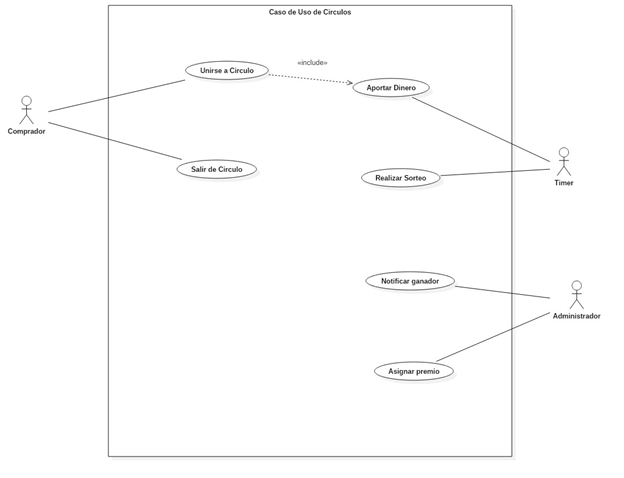
* **Módulo Comunicaciones**
* - Seguir usuario
* - Dejar de seguir usuario
* - Compartir productos
* - Compartir usuarios
* - Enviar mensajes
* - Leer mensajes
* - Eliminar mensajes
* - Reportar usuarios
* - Bloquear usuario
* - Obtener información de redes sociales para ofertas (solo owners)

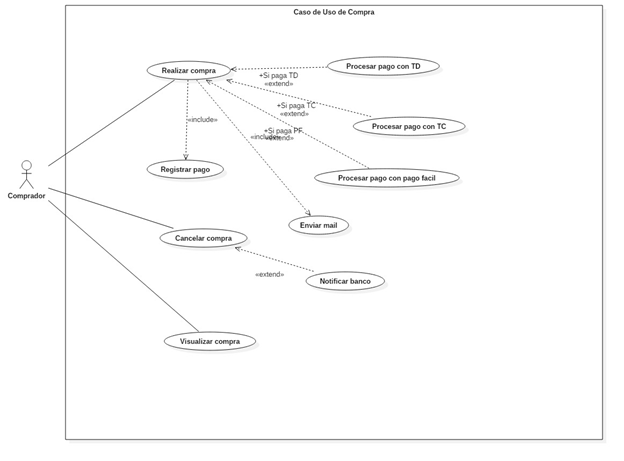
# Vista de Casos de Uso

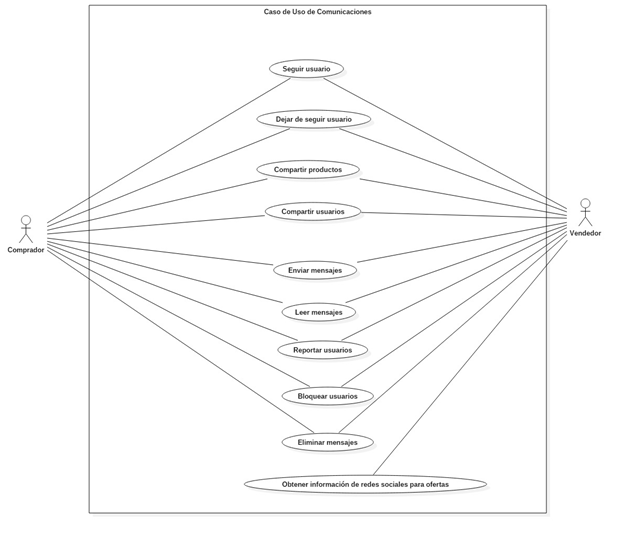
A continuación listamos las funcionalidades que ofrece el sistema, representadas a través de casos de uso que están agrupados según el alcance que tendrá el sistema.

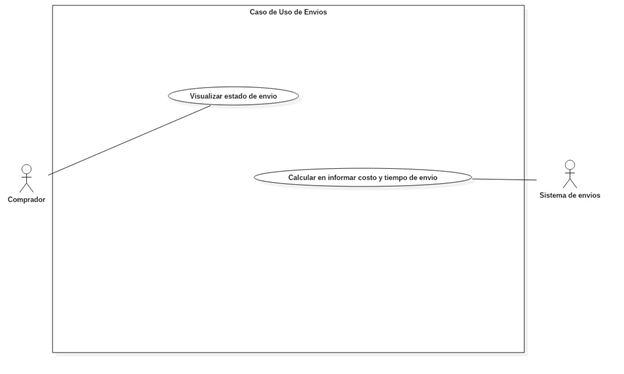


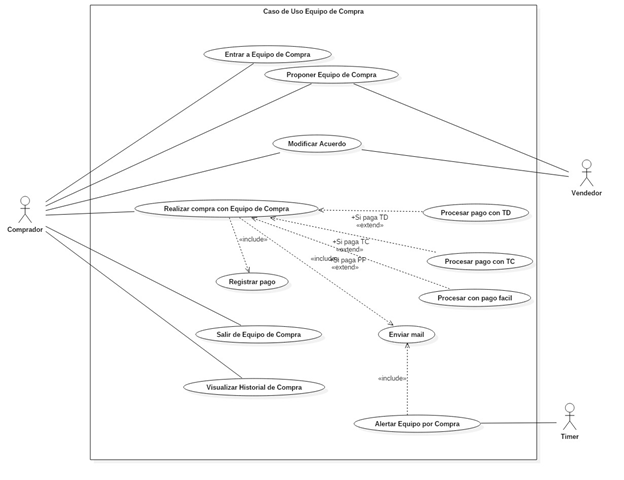


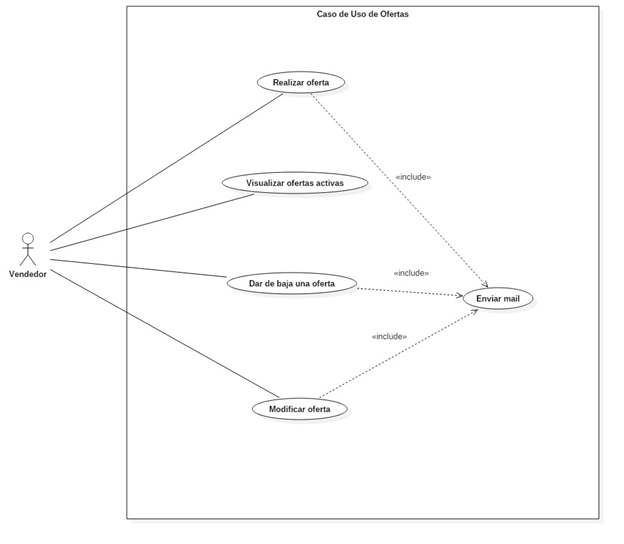


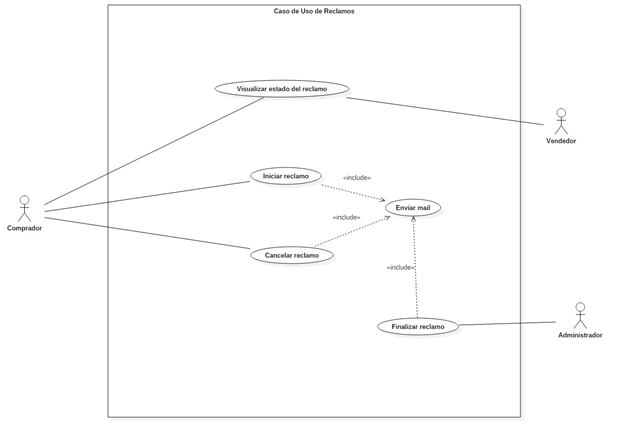


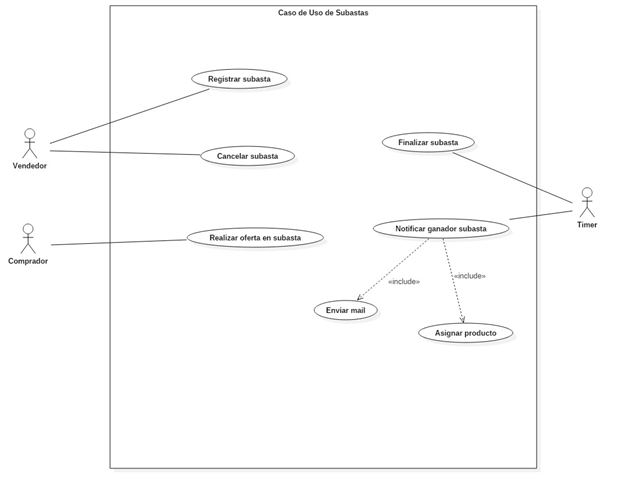


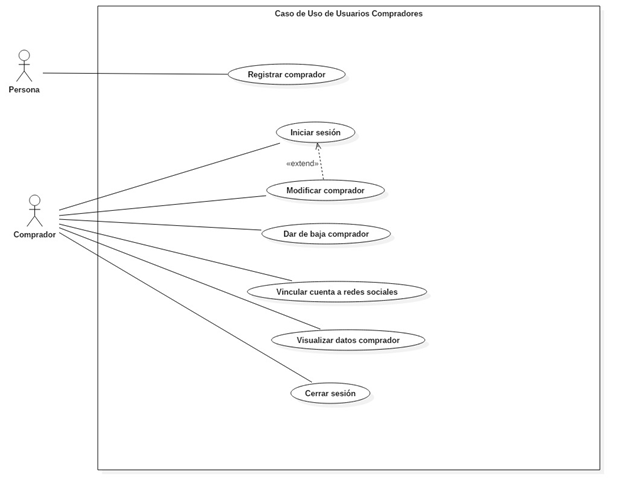


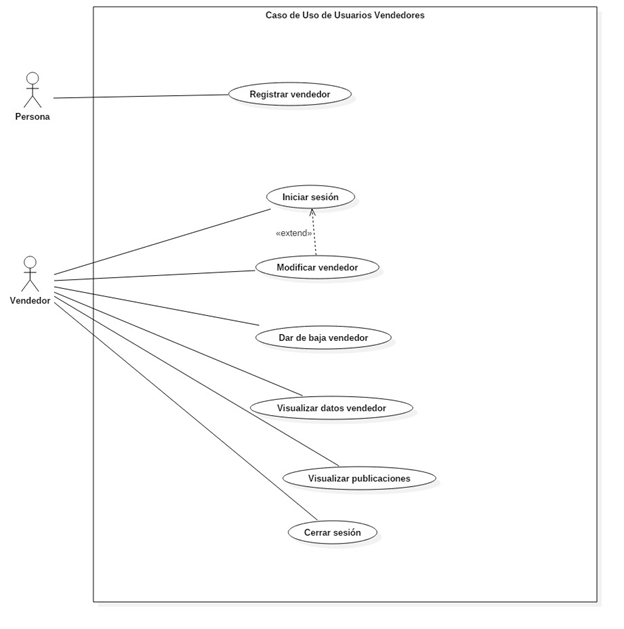








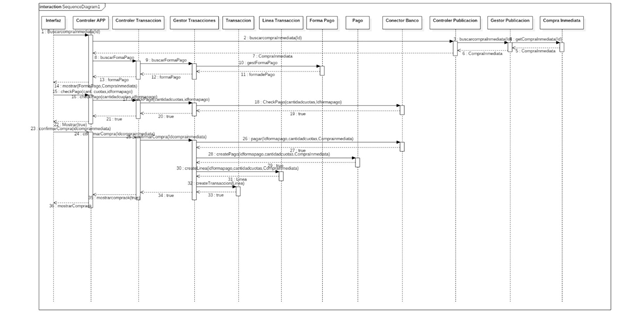




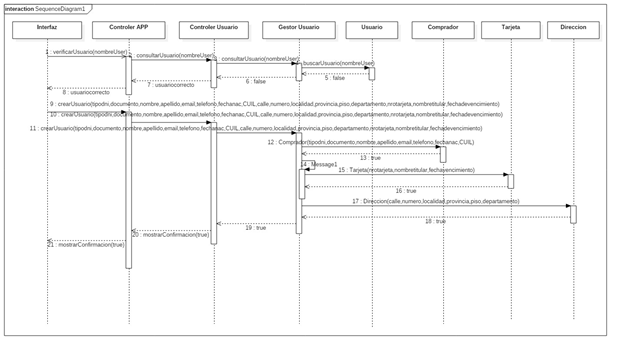
## Realizaciones de los Casos de Uso

A continuación presentamos una serie de escenarios significativos del sistema e ilustramos cómo se realizan, indicando para cada caso cuales son las clases de diseño que participan y cómo interactúan para lograrlo.

Escenario 1: Compra inmediata



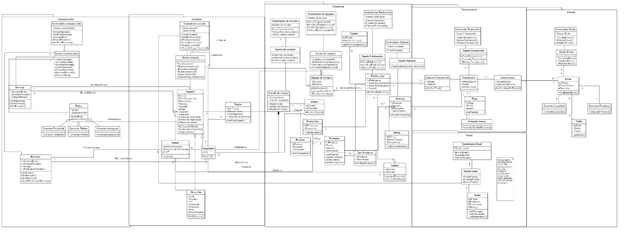
Escenario 2: Registrar comprador



# Vista Lógica

## Visión General

En esta sección presentamos la composición del sistema desde el punto de vista de las clases del diseño.

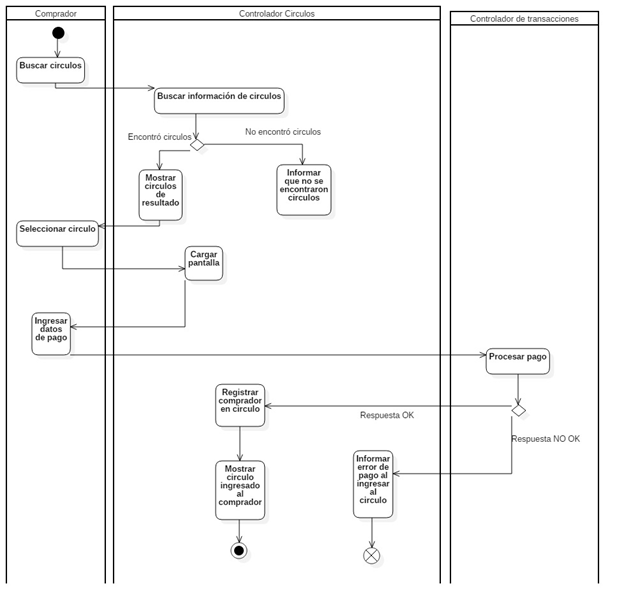


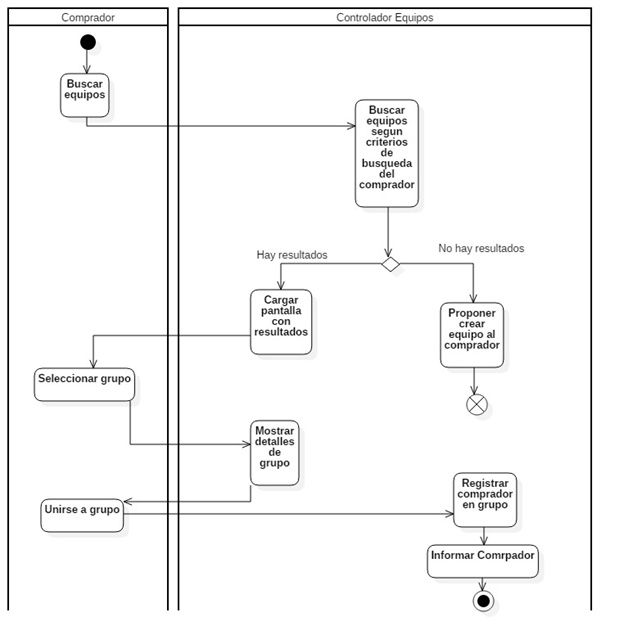
## Paquetes de Diseño Significativos

*N/A*

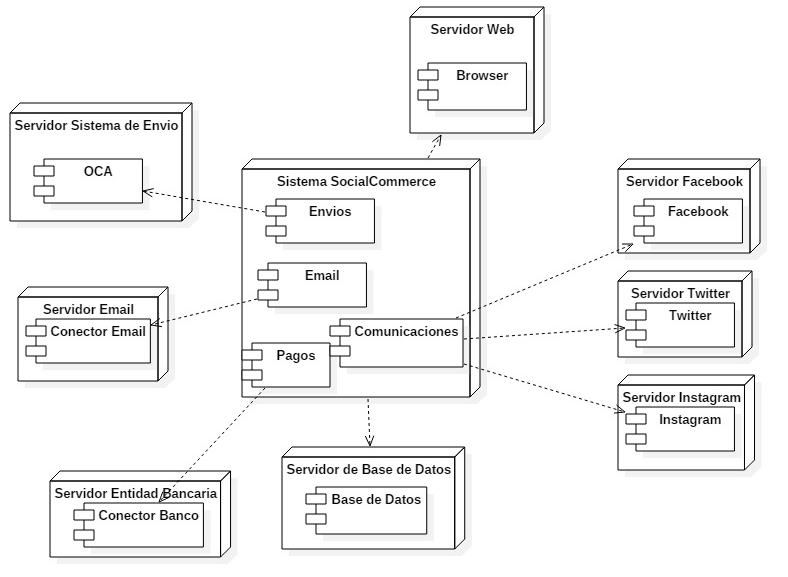
# Vista de Procesos

Algunos procesos del sistema necesitan ser descritos en detalle para dar una idea clara y concisa de cómo deber ser llevados a la práctica.





# Vista de Despliegue



# Vista de Implementación

# 

## Visión General

N/A

## Capas

N/A

# Vista de Datos (opcional)

N/A

# Tamaños y Performance

La base de datos debe ser no relacional dado su performance a la hora de obtener los datos (Por ejemplo: Cassandra)

La webapp debe adaptarse a la resolución en la que es visitada

La webapp debe poder ejecutarse en cualquier sistema operativo con acceso a internet y un explorador sea Chrome 30 en adelante o Firefox 36 en adelante.

# Calidad

Se destaca la estabilidad, fiabilidad, confiabilidad, y disponibilidad que debe poseer el sistema dado la criticidad de los datos que se manejan como también la expectativa del usuario que desee utilizar el sistema en cualquier momento del día.

# Se utiliza el modelo MVC para la construcción de la webapp dada su modularidad y facilidad a los cambios.

El usuario deberá conocer el resultado de sus acciones mediante mensajes de confirmaciones o mensajes detallando el error u omisión del mismo.

La interfaz deberá estar diseñada de tal manera que guíe al usuario en todo momento de la navegación así como también no sobrecargar al usuario con información y darle la sensación de que tiene el control.