<<HARINAS Y GRANOS VALENCIA>>

**Documento de Especificación de Arquitectura**

Realizado por:

Ederson Ramirez

Maxiel Martinez

Wirley Valencia

Ivan Rene Jimenez

***HISTORIAL DE REVISIONES***

| **Fecha** | **Versión** | **Autor** | **Descripción** | **Revisado Por** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

**Contenido**

[1.](#_3znysh7) Documento de Arquitectura de Software 4

[1.1.](#_2et92p0) Introducción 4

[1.2.](#_tyjcwt) Propósito 4

[1.3.](#_3dy6vkm) Alcance 4

[1.4.](#_1t3h5sf) Referencias 4

[1.5.](#_4d34og8) Definiciones acrónimos y abreviaciones 4

[2.](#_2s8eyo1) Generalidades del Proyecto 5

[2.1.](#_17dp8vu) Problema a Resolver 5

[2.2.](#_3rdcrjn) Descripción General del Sistema a Desarrollar 5

[2.3.](#_26in1rg) Identificación de los Stakeholders y sus responsabilidades 5

[3.](#_lnxbz9) Vistas de la arquitectura 5

[3.1.](#_35nkun2) Vista de Casos de Uso (Se debe colocar cada uno de los casos de uso que han sido elaborados y requeridos para el S.I.) 5

[3.2.](#_1ksv4uv) VISTA DE PROCESOS 5

[3.3.](#_44sinio) VISTA LÓGICA 5

[3.4.](#_z337ya) VISTA DE IMPLEMENTACIÓN 6

[3.5.](#_3j2qqm3) VISTA DE DESPLIEGUE 6

[4.](#_1y810tw) Arquitectura en capas 6

[5.](#_4i7ojhp) VISTA DE DATOS 6

[5.1.](#_2xcytpi) Modelo Relacional normalizado tercera forma normal de la base de datos. 6

[6.](#_1ci93xb) Definición de Interfaces de Usuario 6

[7.](#_3whwml4) Características Generales de Calidad 6

[7.1.](#_2bn6wsx) Tamaño y performance 6

[7.2.](#_1pxezwc) Calidad 6

[7.3.](#_49x2ik5) Usabilidad 7

[7.4.](#_2p2csry) Eficiencia 7

[7.5.](#_147n2zr) Seguridad 7

[7.6.](#_3o7alnk) Confiabilidad 7

[7.7.](#_23ckvvd) Mantenimiento 7

[7.8.](#_ihv636) Estándares (Normas de calidad): Mencionar dos o tres normas, y argumentar cómo esas normas se aplican para el sistema de información. 7

# Documento de Arquitectura de Software

## Introducción

Este documento proporciona una visión detallada de la arquitectura de software que se utilizará en la tienda de abarrotes para manejar sus operaciones diarias y mejorar la eficiencia del negocio. La arquitectura de software es un aspecto crítico para cualquier empresa que desee mantenerse competitiva en el mercado actual, y la tienda de abarrotes no es una excepción. Este documento describe la estructura y los componentes del software, así como su funcionamiento en conjunto para lograr los objetivos de la tienda de abarrotes. La implementación de esta arquitectura de software permitirá a la tienda de abarrotes mejorar su capacidad para procesar pedidos, administrar inventarios y ofrecer una experiencia de compra más atractiva para los clientes.

## Propósito

El propósito de este documento es definir la estructura y el diseño del software que se utilizará para crear y mantener el sistema de información de la tienda de abarrotes en línea Harinas y Granos Valencia. Este documento servirá como guía para los desarrolladores y otros miembros del equipo de desarrollo, proporcionando una visión general clara y coherente de la arquitectura y el diseño del sistema. El documento también establecerá las decisiones clave de diseño y arquitectura, incluyendo la selección de tecnologías, la descripción de los componentes del sistema, la interacción entre estos componentes y las decisiones de integración con otros sistemas y servicios externos.

## Alcance

* El sistema de información debe incluir todas las medidas de calidad y seguridad, para satisfacer los requerimientos del usuario.
* El sistema de información debe permitir gestionar la ventas, así como registrar ventas y ver informe de ventas realizadas
* El sistema debe permitir gestionar la compras y los gastos, así como registrar las compras y los gastos de la microempresa.
* El sistema de información debe permitir llevar un registro de los productos que se tienen en el inventario.
* El sistema de información debe permitir llevar un registro de los pedidos realizados y además notificar a las partes cuando un pedido esté listo o cuando se ingrese un pedido nuevo.

## Referencias

1. Documento de Especificación de Requerimientos no funcionales.
2. Documento de Visión del Proyecto.
3. Plan de Proyecto del Sistema

## Definiciones acrónimos y abreviaciones

**Arquitectura de software:**

La arquitectura de software es la estructura fundamental y organización de un sistema de software que establece cómo sus componentes trabajan juntos para lograr sus objetivos, y es importante para el diseño, desarrollo, implementación y mantenimiento de un sistema de software.

**Descripción de arquitectura:**

La arquitectura es la planificación, diseño y construcción de estructuras físicas y organizacionales para satisfacer las necesidades humanas. La arquitectura implica la consideración de factores como la función, la forma, la estética, la sostenibilidad, la eficiencia, la seguridad y la accesibilidad.

**Vistas:**

Una vista en la arquitectura de software es una representación específica del sistema que se enfoca en un aspecto particular de la arquitectura, como componentes, despliegue, procesos, datos, seguridad, rendimiento o casos de uso. Las vistas se utilizan para ayudar a los desarrolladores y otros interesados a entender la estructura y organización del sistema desde diferentes perspectivas y niveles de abstracción.

**Tipos de vistas:**

**Vista de arquitectura:** Es una representación global de la estructura y organización de un sistema de software. Muestra los componentes principales del sistema, sus interacciones y cómo se relacionan con los objetivos del negocio.

**Vista de casos de uso:** se enfoca en los casos de uso del sistema y cómo se relacionan con los componentes y procesos del sistema.

**Vista** de procesos: se enfoca en los procesos y servicios que se ejecutan en el sistema.

**Vista de lógica:** La vista de lógica se enfoca en cómo el software procesa los datos y las acciones dentro del sistema. Esta vista describe la estructura y organización de los componentes y módulos de software, y cómo interactúan para realizar tareas específicas.

**Vista de implementación:** Muestra cómo se implementan los componentes del software en el sistema operativo, el servidor de aplicaciones y otras tecnologías de infraestructura. Esta vista describe las relaciones entre los componentes del software y los recursos físicos necesarios para su implementación, como servidores, bases de datos y sistemas de almacenamiento.

**Vista de despliegue:** se enfoca en cómo se distribuye el software en diferentes máquinas o dispositivos.

**Vista de datos:** se enfoca en los datos y su flujo a través del sistema.

**Stakeholders:**

Un stakeholder es cualquier individuo o grupo que tiene un interés o una influencia en un proyecto, una empresa o una organización.

**Caso de uso:**

Un caso de uso es la descripción de una acción o una actividad. Un diagrama de caso de uso es una descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso.

**Actores:**

Los actores en un ml son clases con el estereotipo <<actor>> y tienen un estereotipo icono estándar. Nombre de la clase es el nombre del actor. Una clase actor puede tener atributos y comportamiento. Los actores pueden tener las mismas relaciones qué las clases.

**Clase:**

Grupo de elementos de un conjunto que tiene características comunes.

**Atributo:**

Atributos son características individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades.

**Método:**

Método es una porción de código que realiza tareas asociadas a un objeto, básicamente un método es una función que pertenece a un objeto o clase.

**Diagrama de actividades:**

Un diagrama de actividades es un diagrama de flujo que muestra actividades ejecutadas por un sistema.

**Diagrama de clases:**

Un Diagrama de clases es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones y las relaciones entre los objetos.

**Uml:**

El lenguaje unificado de modelado es una forma de mostrar visualmente el comportamiento y la estructura de un sistema o proceso.

# Generalidades del Proyecto

## Problema a Resolver

La tienda Harinas y Granos Valencia no proporciona una experiencia de usuario óptima, ya que no tiene sus procesos digitalizados, lo que causa que la información se pierda, no esté disponible o que se tarde mucho en hacer la contabilidad. lo que resulta en un flujo de clientes no demasiado estable, y ha hecho que el crecimiento de la empresa no sea el esperado.

## Descripción General del Sistema a Desarrollar(en general y por módulo).

El sistema de información para la tienda de abarrotes será una plataforma tecnológica diseñada para administrar y gestionar todas las operaciones relacionadas con la venta de productos.

Este sistema de información consta de varios módulos, los cuales se encargan de realizar tareas específicas u operaciones para ayudar a gestionar los distintos rubros dentro de la empresa.

● **Módulo de ventas y pedidos**: Estos módulos permitirán gestionar las ventas de la microempresa H&G Valencia implementando funcionalidades tales como un carrito de compras, facturación y un conteo de las ventas diarias y mensuales, además de darle un mejor servicio a los clientes, también ver los pedidos de los clientes e informarles que su pedido está listo y que pueda pasar por él mostrando el

número de pedido que le fue asignado desde el software.

● **Módulo de compras:** Este sistema de información debe agrupar todos los datos necesarios de manera muy concisa para una correcta administración, registrando en tiempo real las compras, facilitando el proceso tedioso de las compras. dejando atrás el papel y el lápiz que es muy común en estos días.

● **Módulo de inventario:** Este módulo permitirá gestionar el proceso de inventario de Harinas y granos Valencia implementando funcionalidades tales como registrar producto, eliminar productos, editar información del producto y generar informe de inventario. Esto con el fin de optimizar el tiempo que toma el proceso inventario de la microempresa H&G Valencia.

Todos los módulos están interconectados y trabajan juntos para permitir a que la tienda funcione de manera eficiente y efectiva.Además, le permite al administrador al dueño todas las operaciones del negocio, para así tomar decisiones de manera más completa, eficiente y rentable para el bien de la tienda.

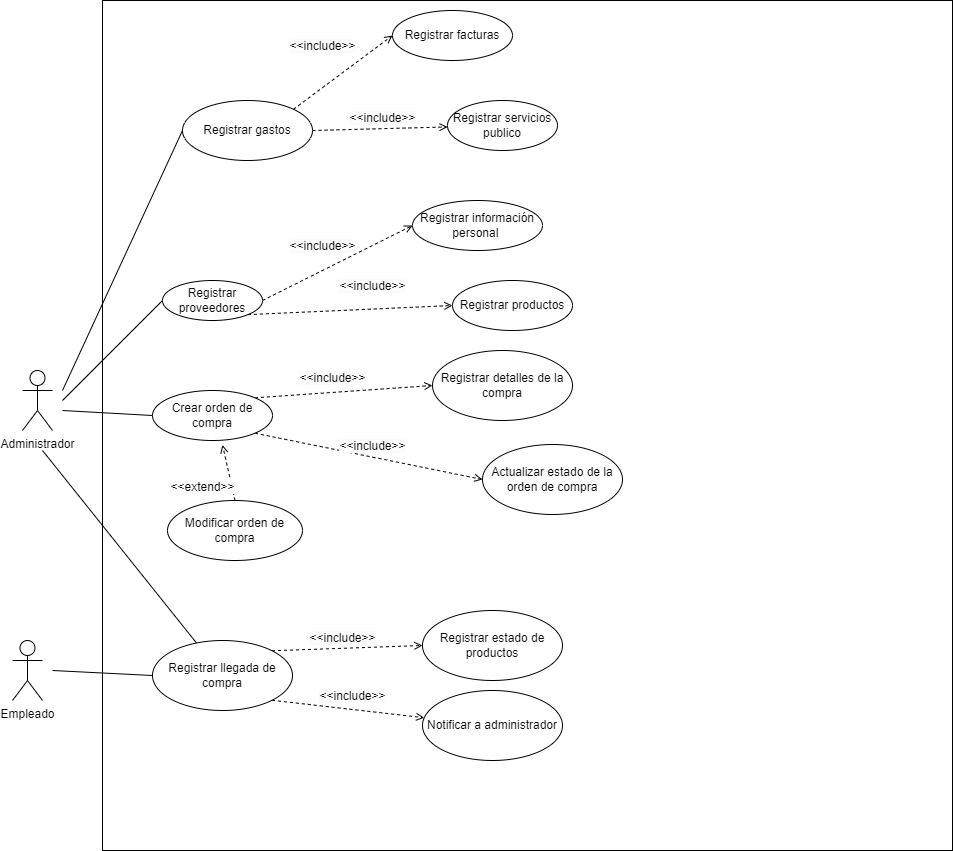
## Identificación de los Stakeholders y sus responsabilidades

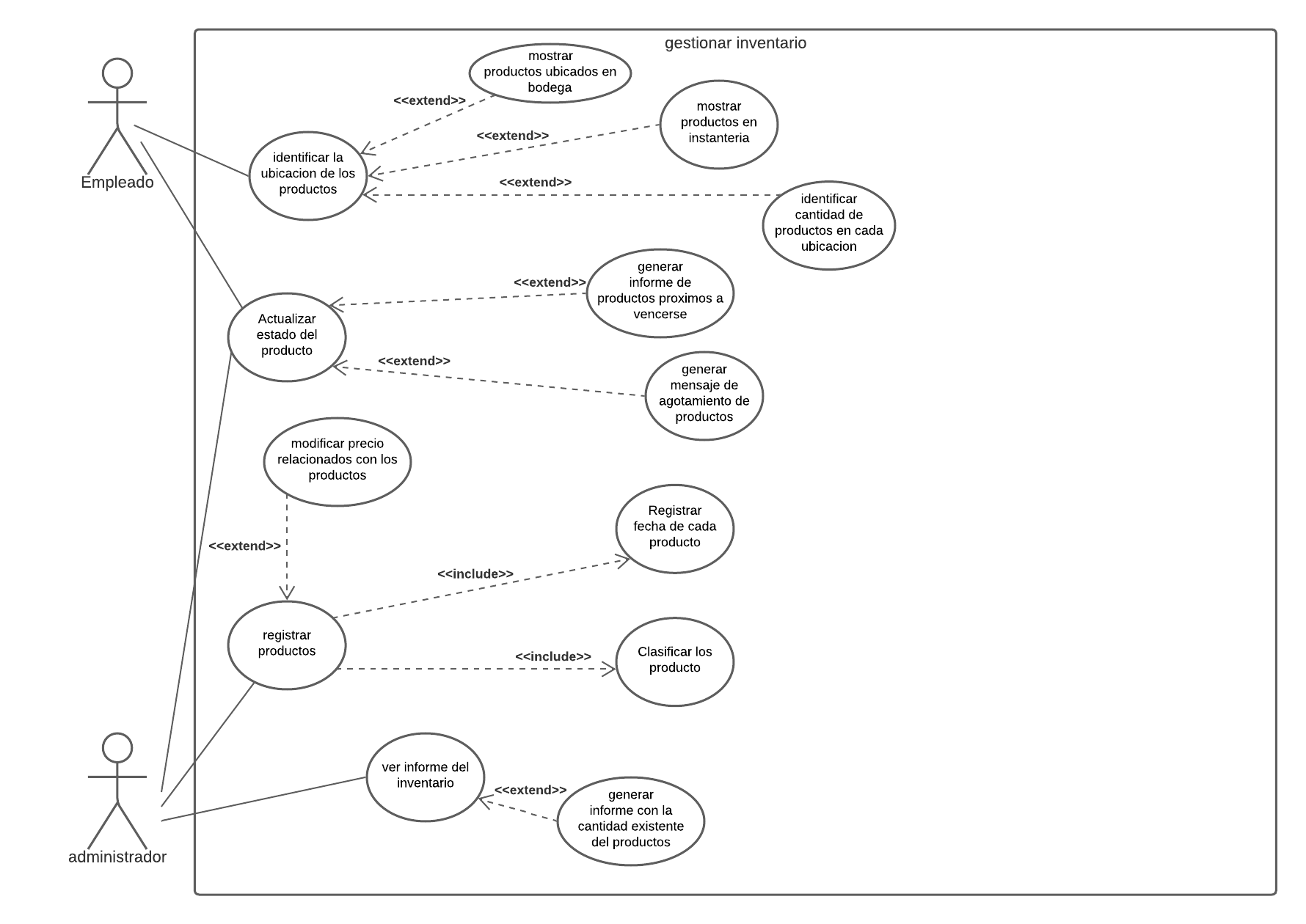
| **STAKEHOLDER** | **DESCRIPCIÓN** | **ESCENARIO** | **Caso de Uso** |
| --- | --- | --- | --- |
| Administrador | Es el usuario dueño de la microempresa . | * Escenario de   negocios   * Escenario de ventas * Escenario de pedidos * Escenario de proveedores | * Generar informe de ventas * Registrar información de los proveedores * Consultar productos faltantes * Generar orden de compra * Admitir Compras * Actualizar estado del producto * Registrar productos * Ver informe del inventario * Verificar pedidos |
| Cliente | Es la persona que interactúa con el sistema para elegir productos  y hacer que los pedidos | * Escenario de ventas * Escenario de pedidos | * Hacer búsqueda de productos * Gestionar pedidos * Pagar productos * gestionar pedidos |
| Empleado | Es el usuario que verifica las ventas y los pagos. | * Escenario de catálogo * Escenario de carrito | * Identificar la ubicación de los productos * Actualizar estado del producto |

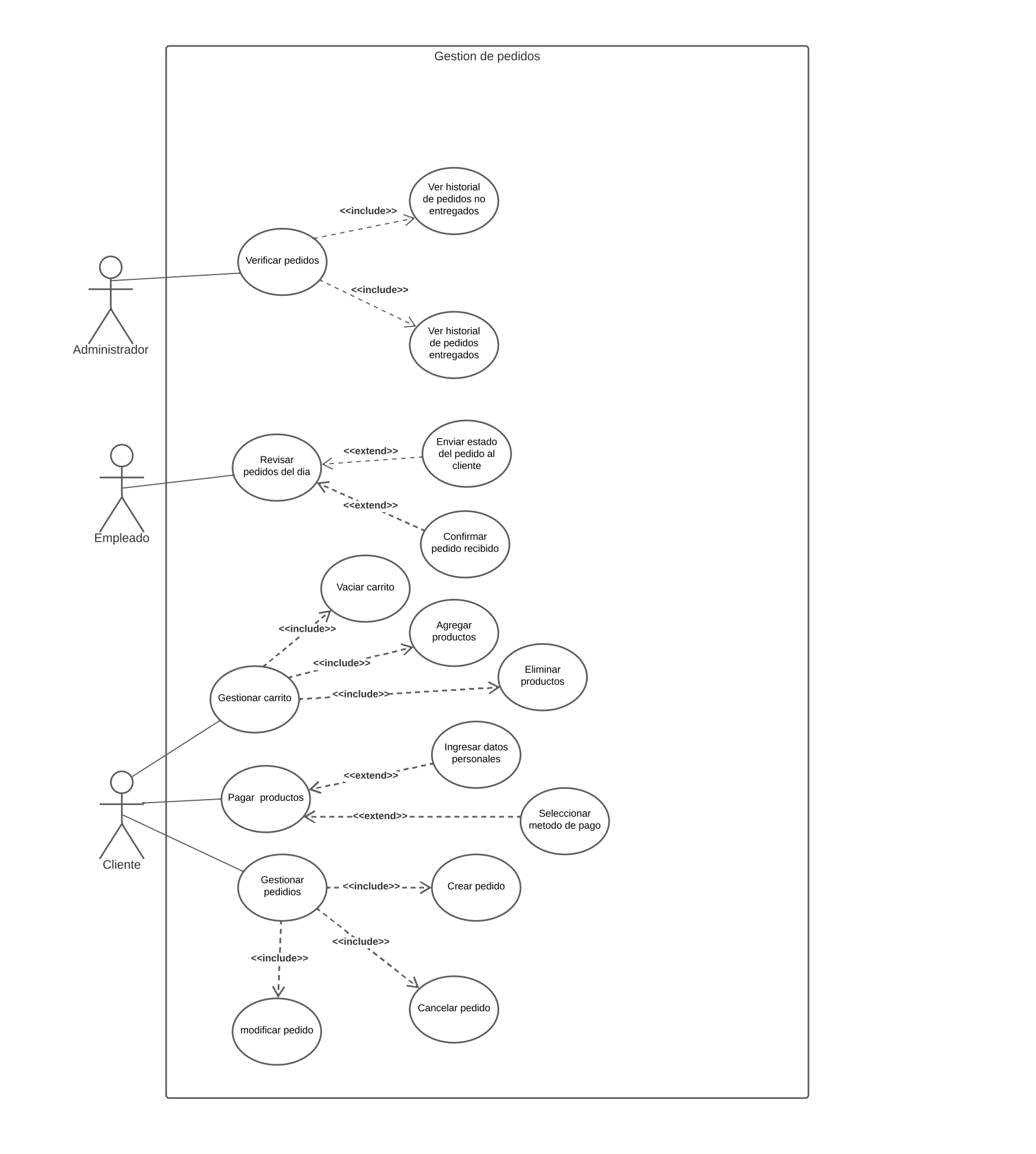
# Vistas de la arquitectura

## Vista de Casos de Uso (Se debe colocar cada uno de los casos de uso que han sido elaborados y requeridos para el software)

****

****

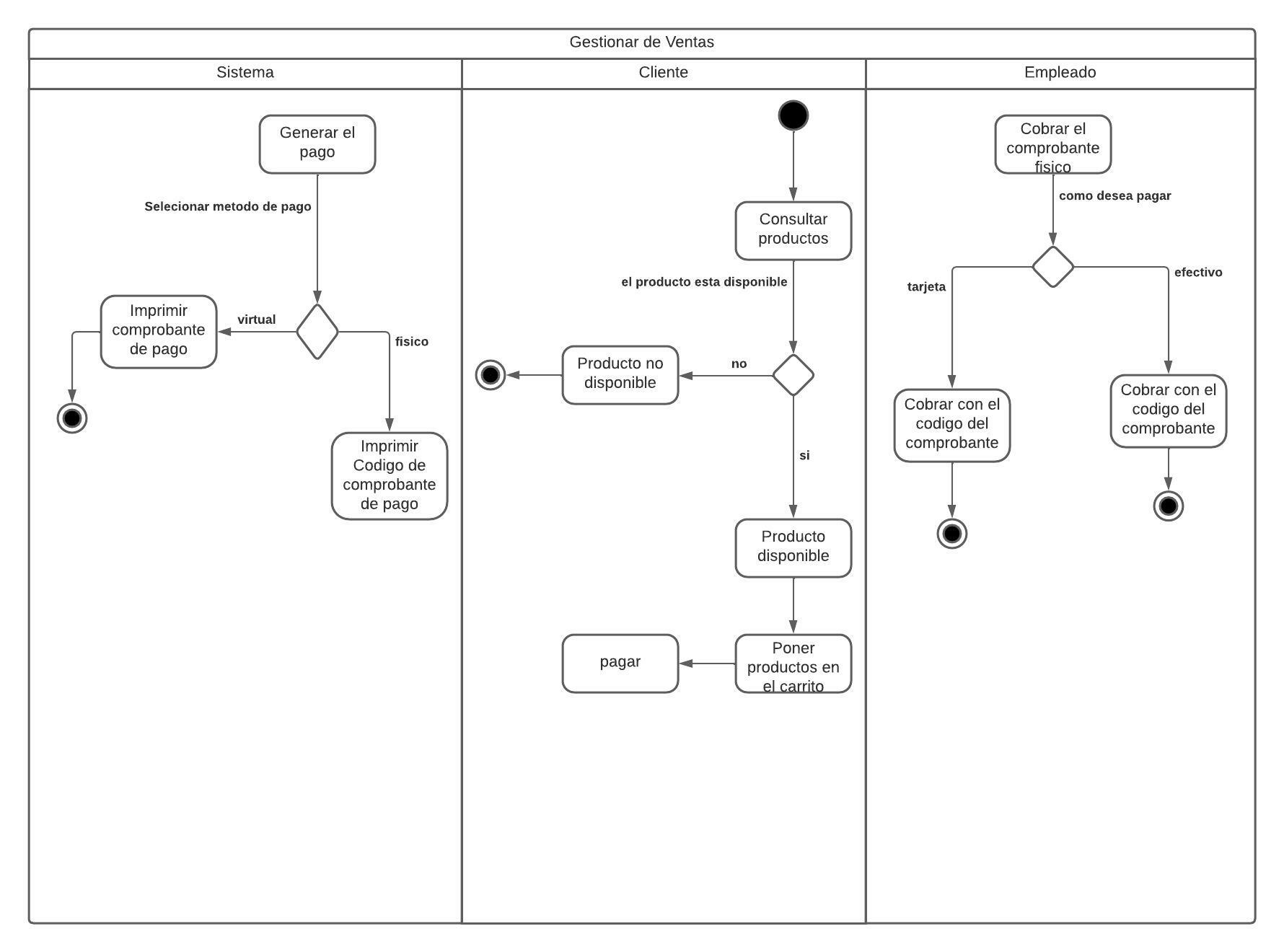
****

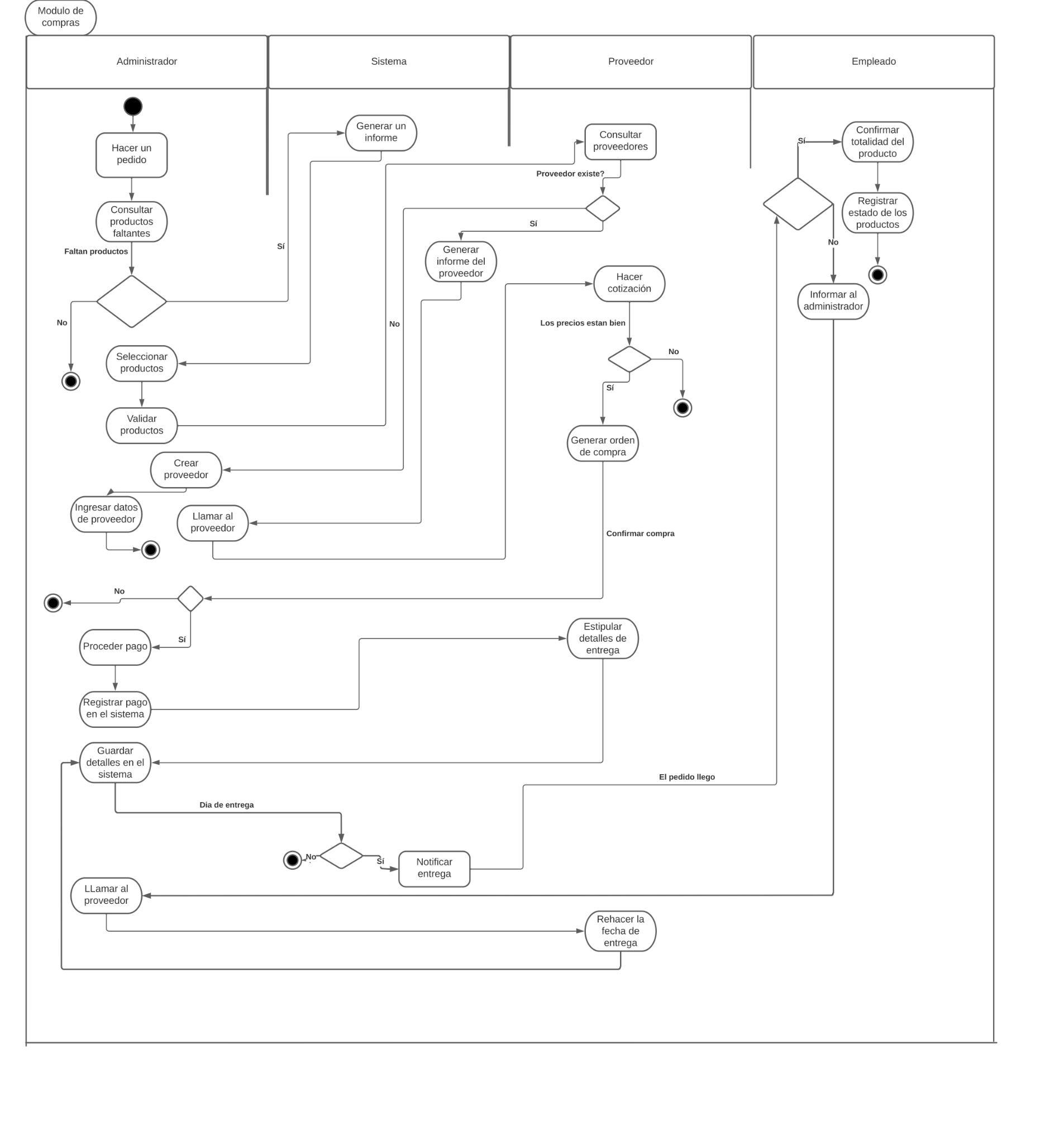
****

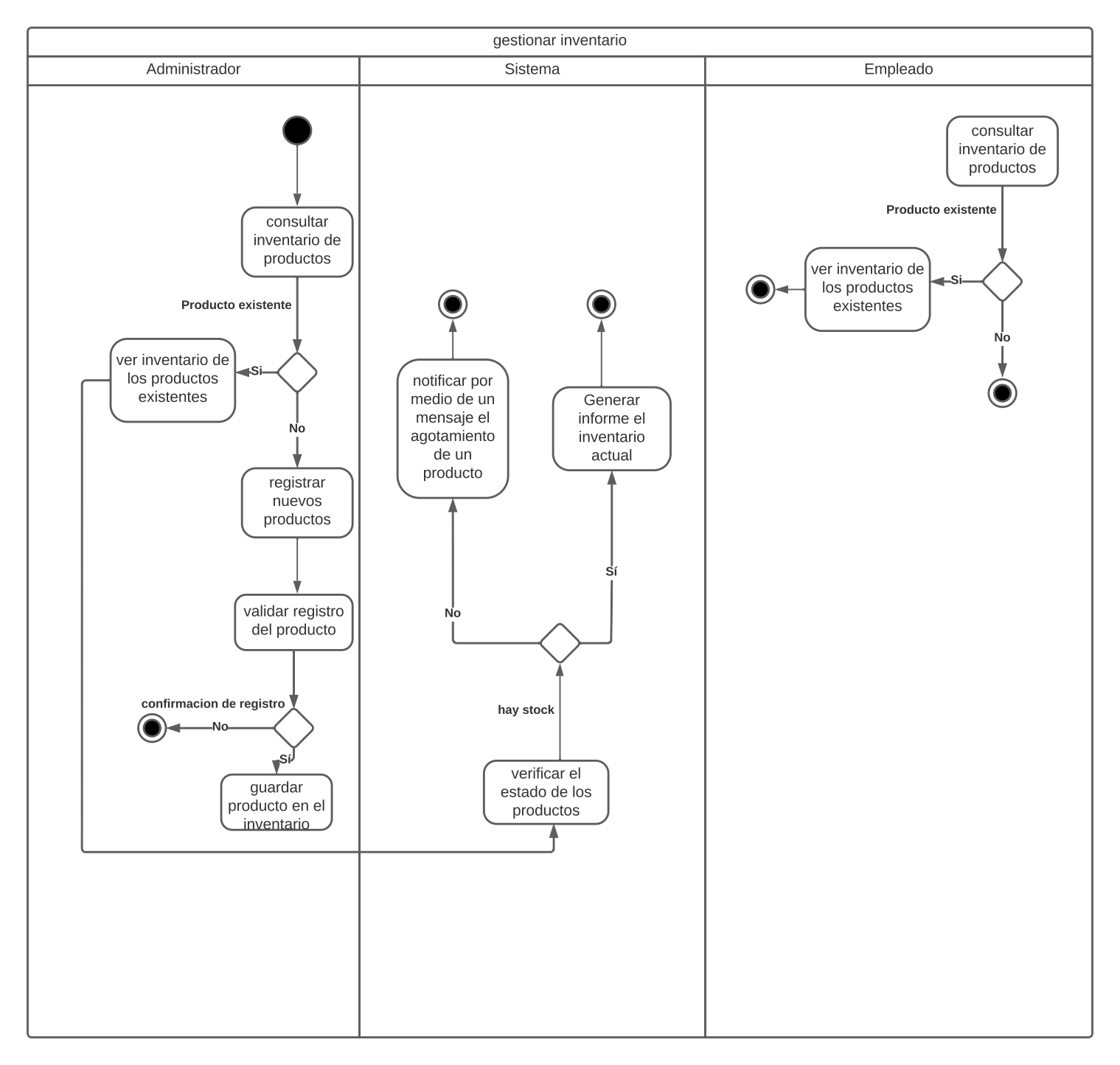
## VISTA DE PROCESOS

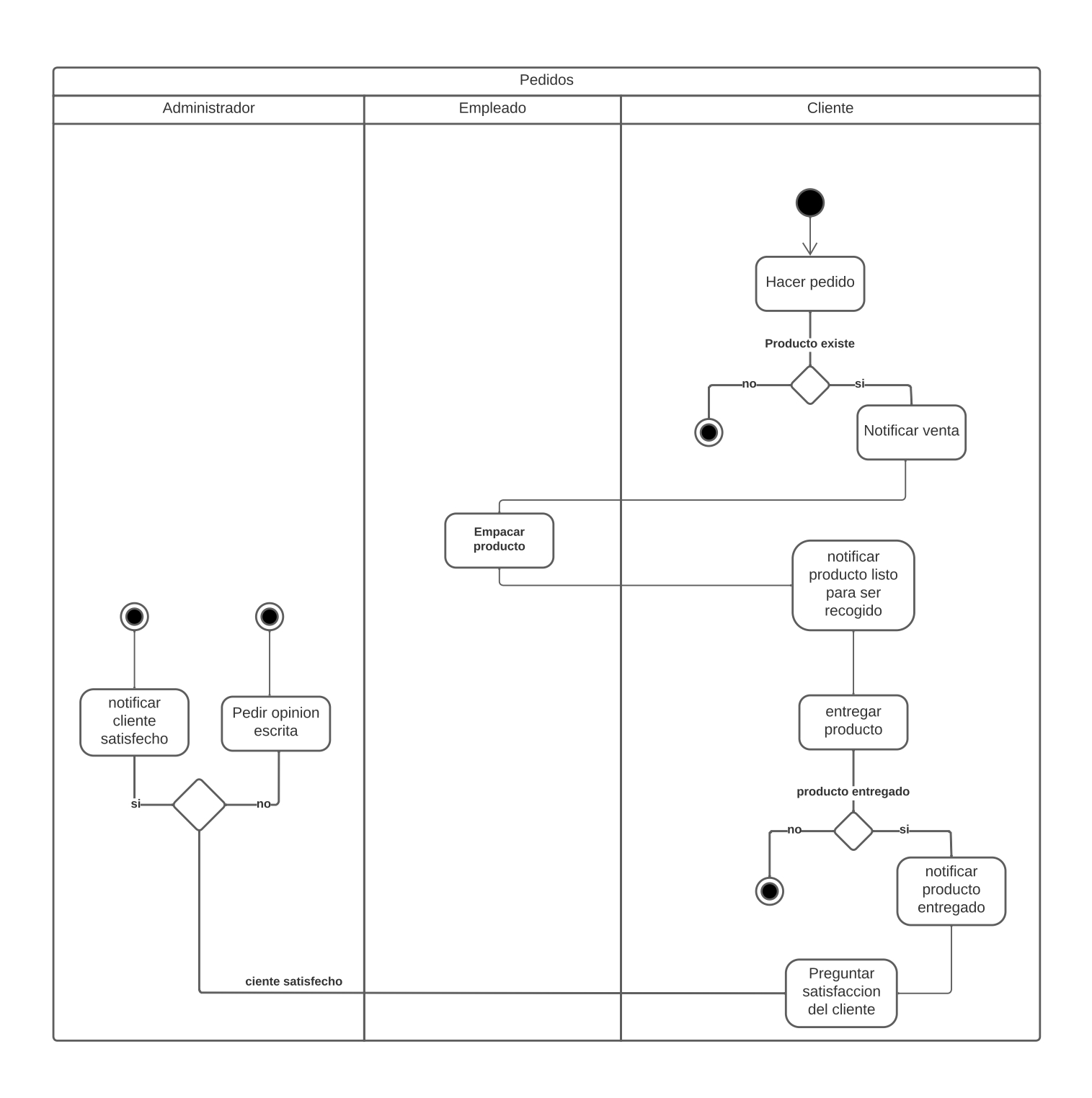
Todos los diagramas que hayan elaborado en fase II, por módulo (aplica para los puntos 3.2.1 al 3.3.2), debidamente argumentados cada uno de los diagramas

### Diagrama de Actividades

****

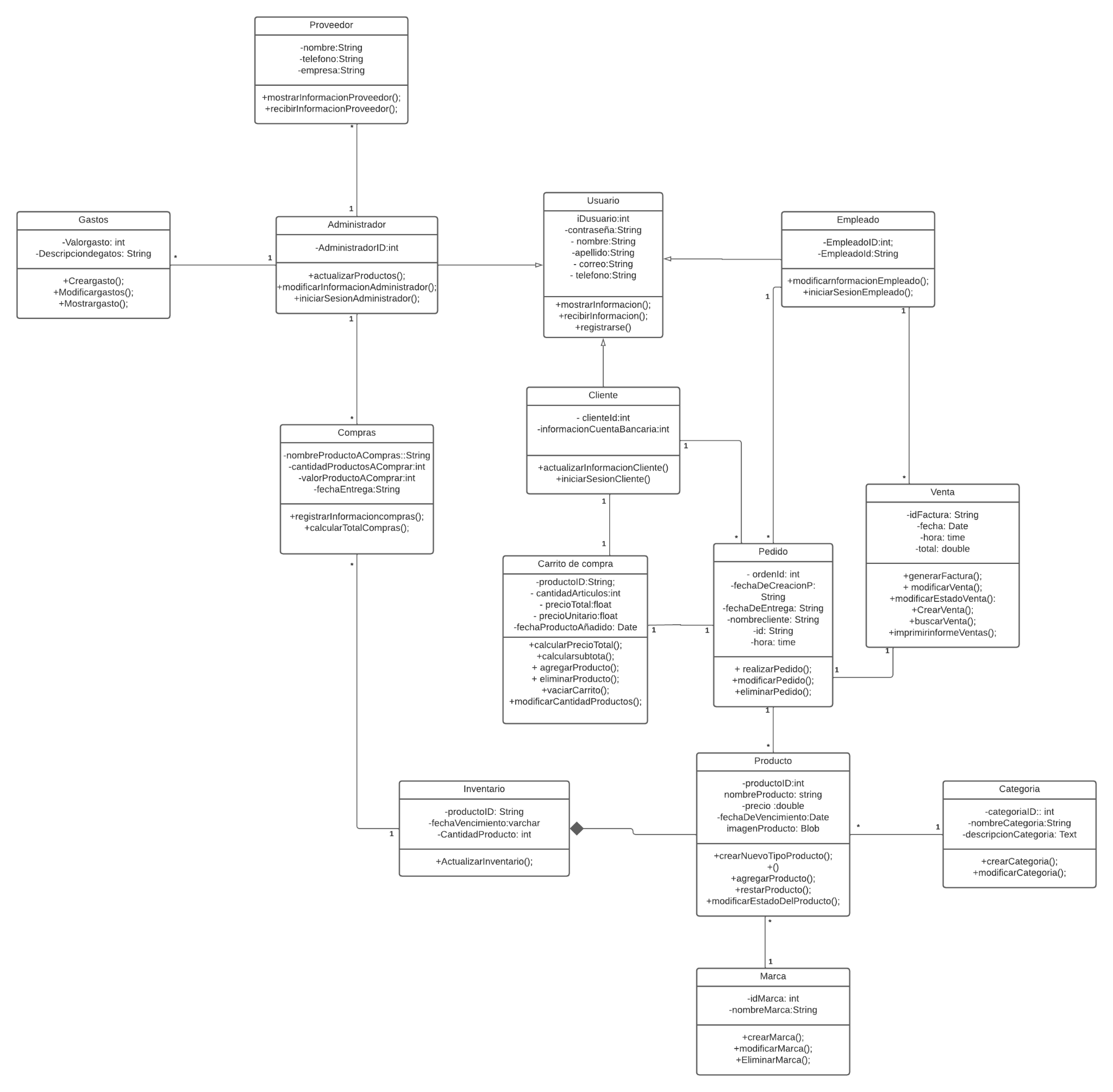
****

****

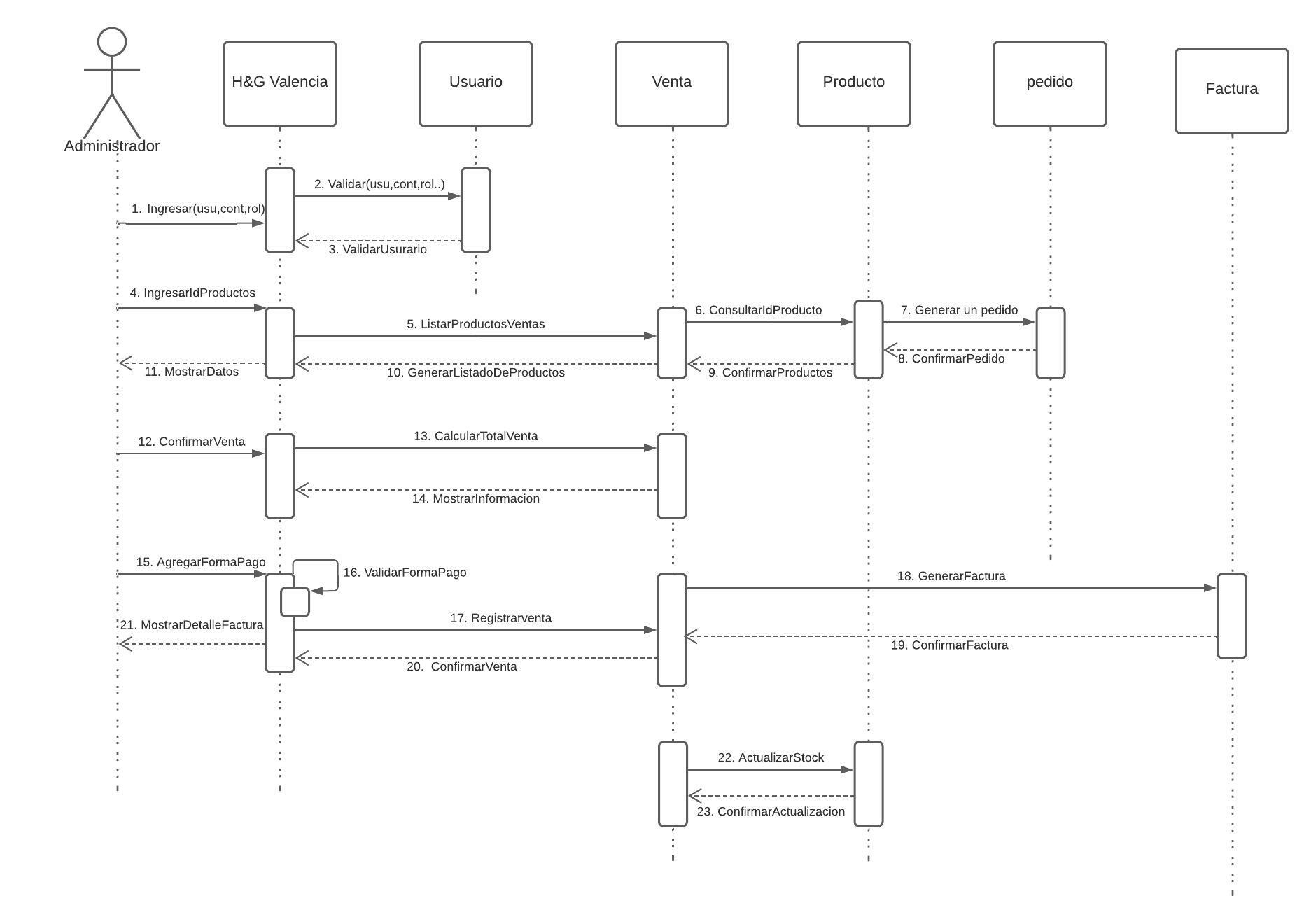
****

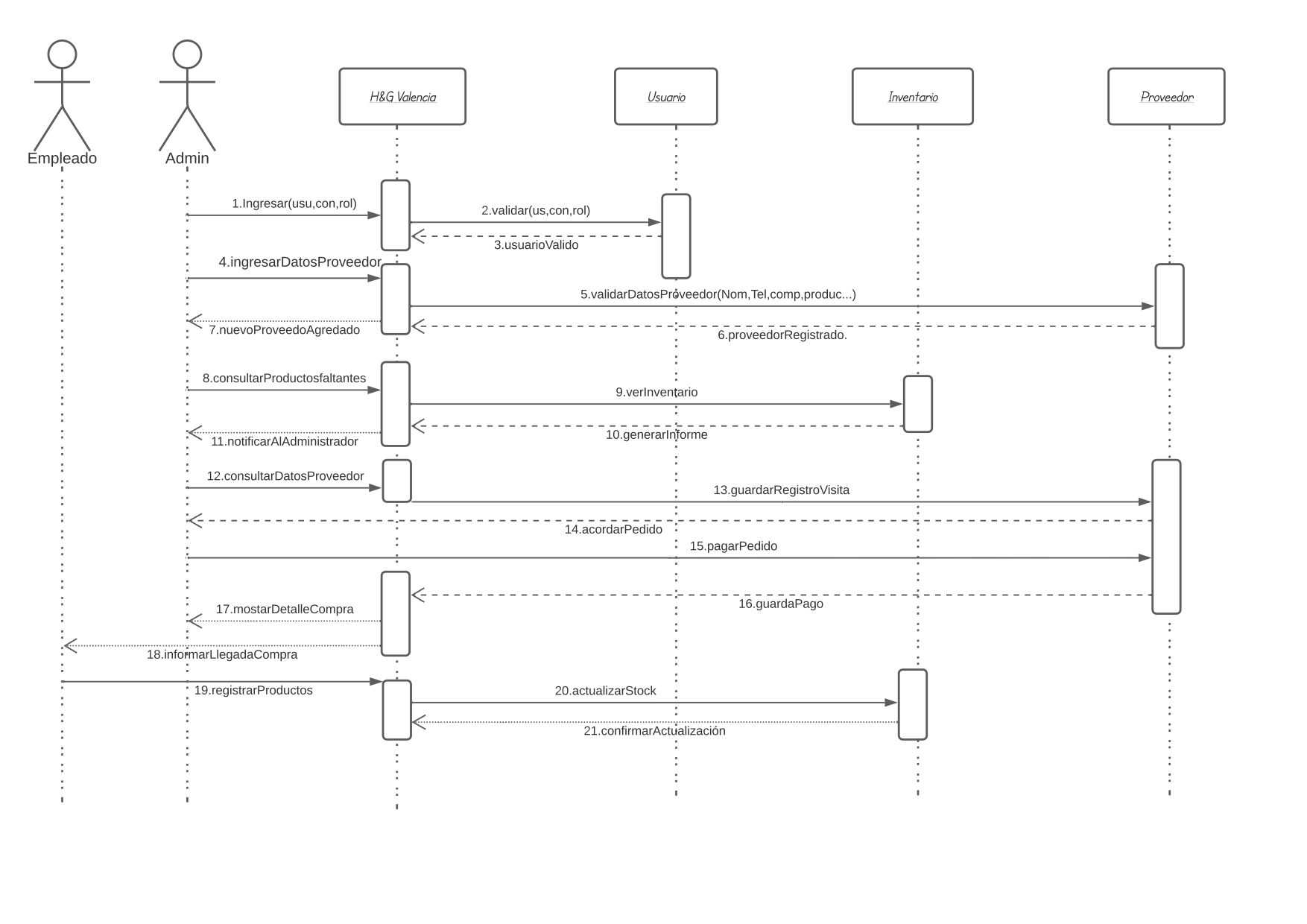
## VISTA LÓGICA

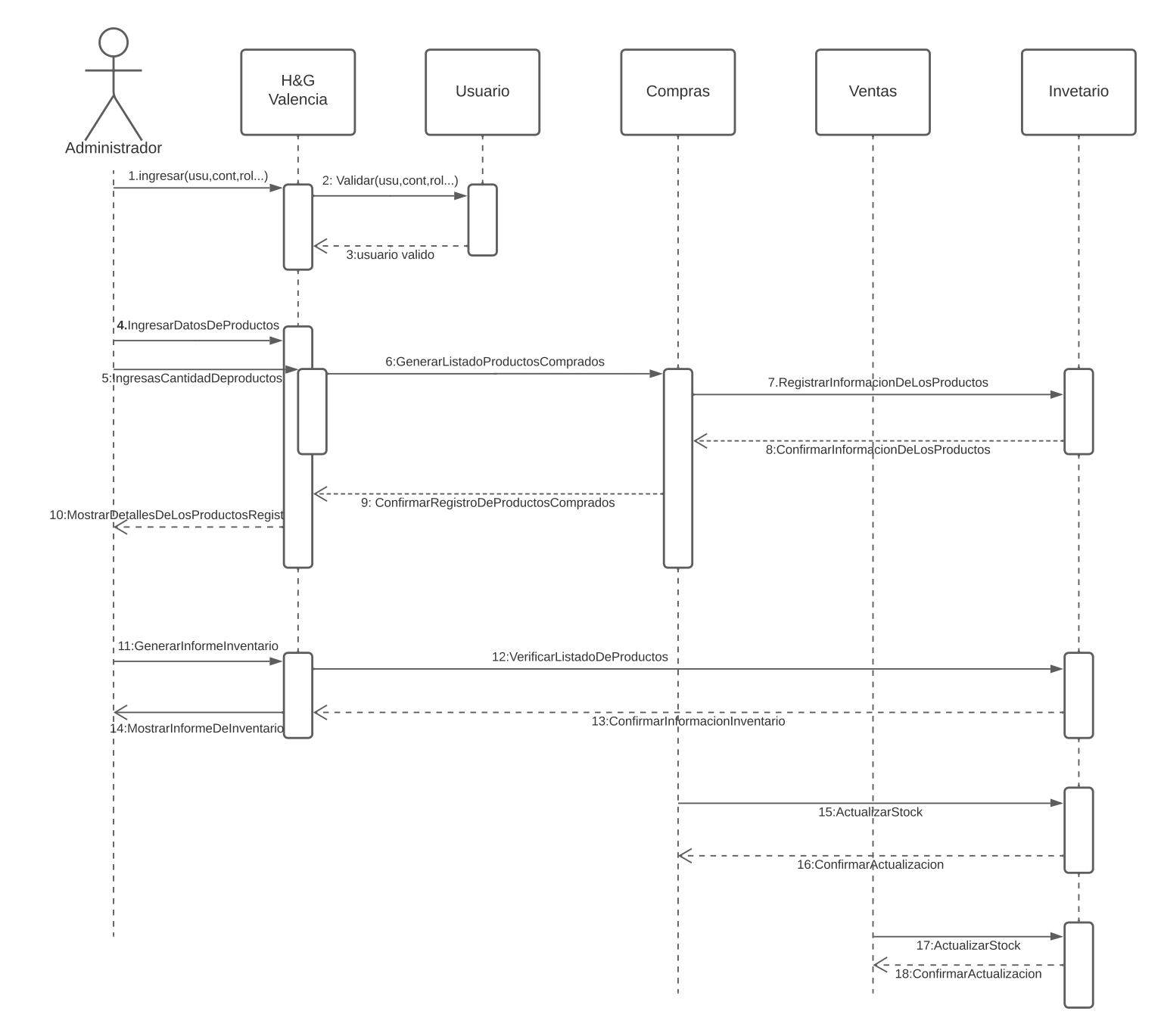
### Diagramas - Clases(Actualizado)

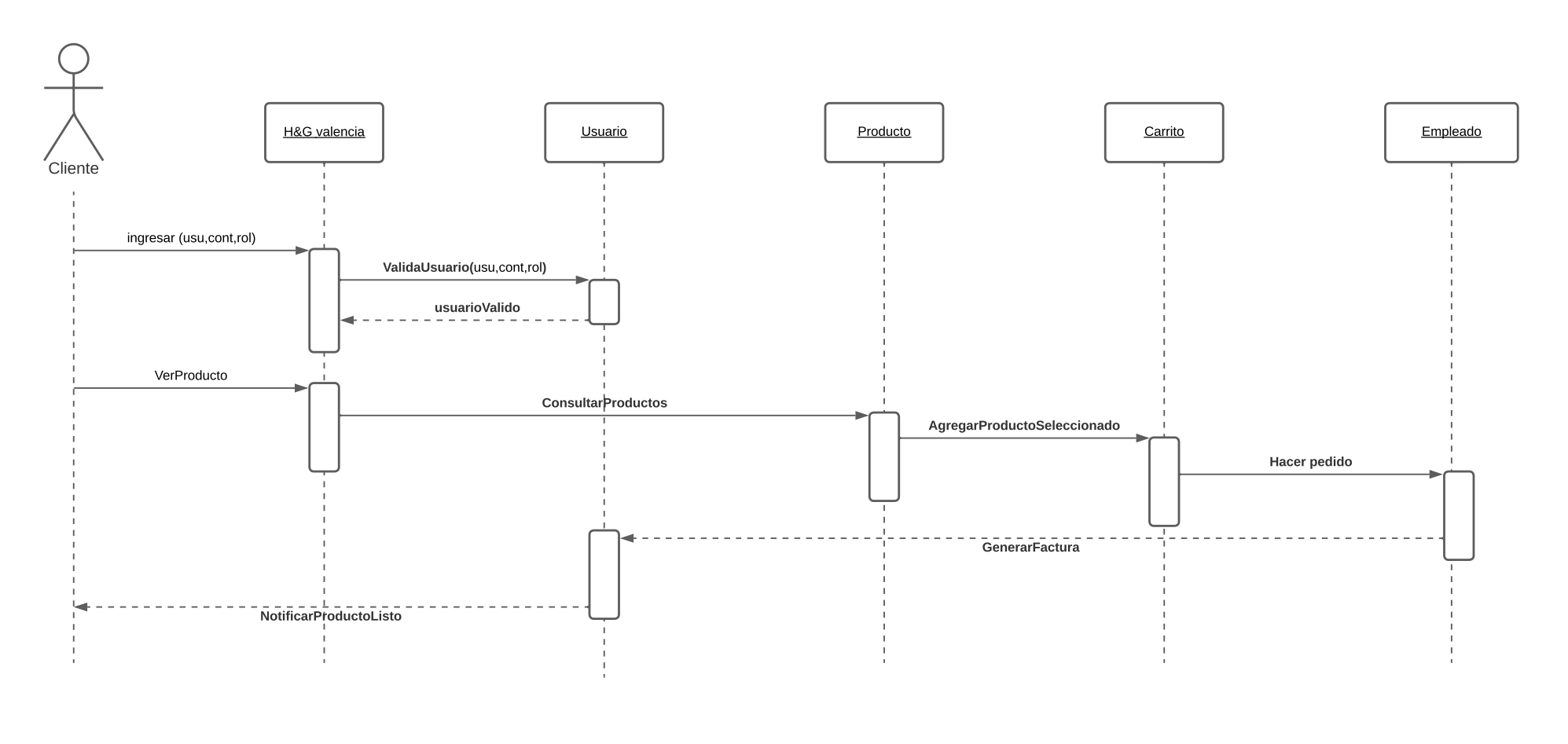
****

### Diagramas - Secuencia

****

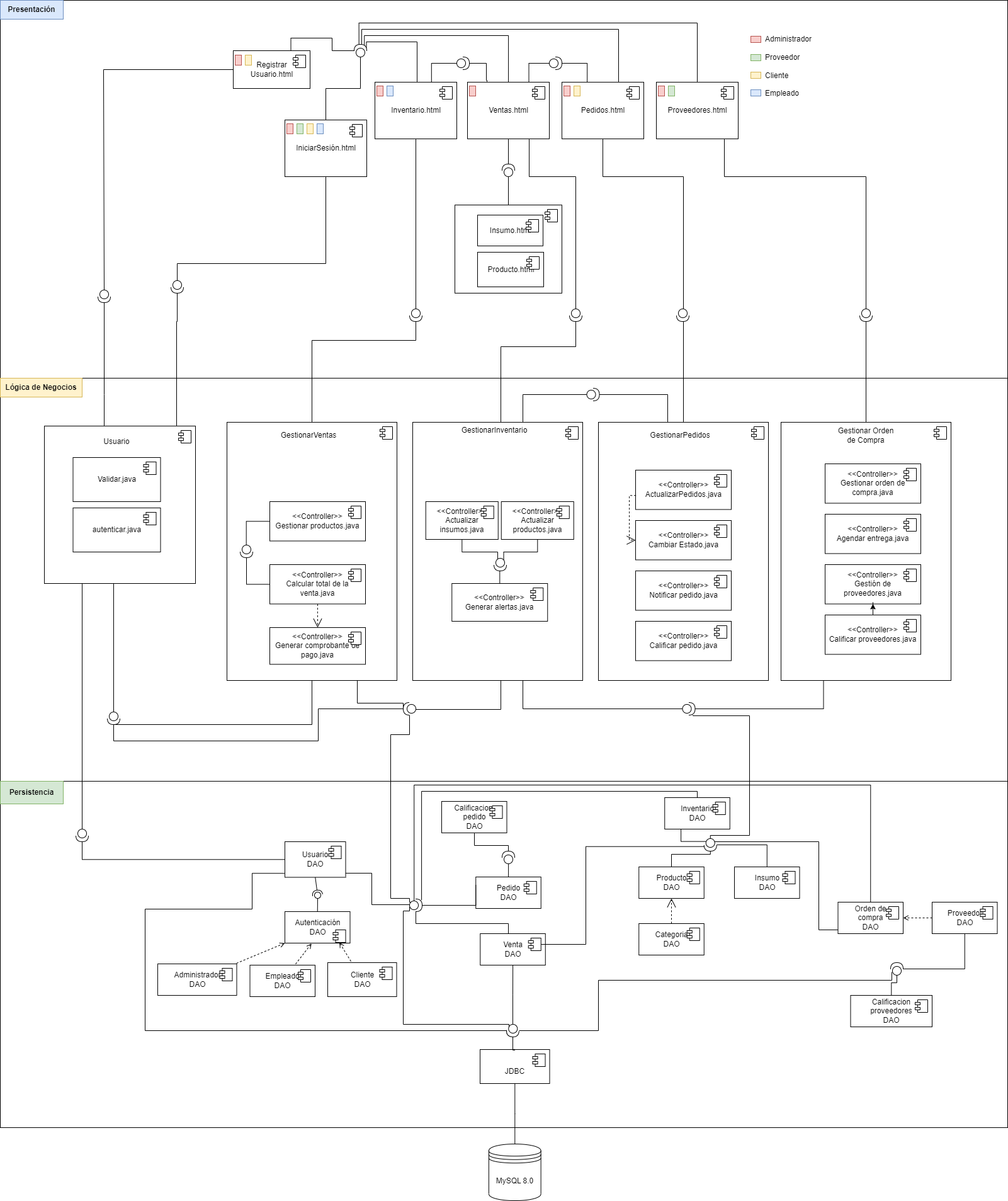
****

****

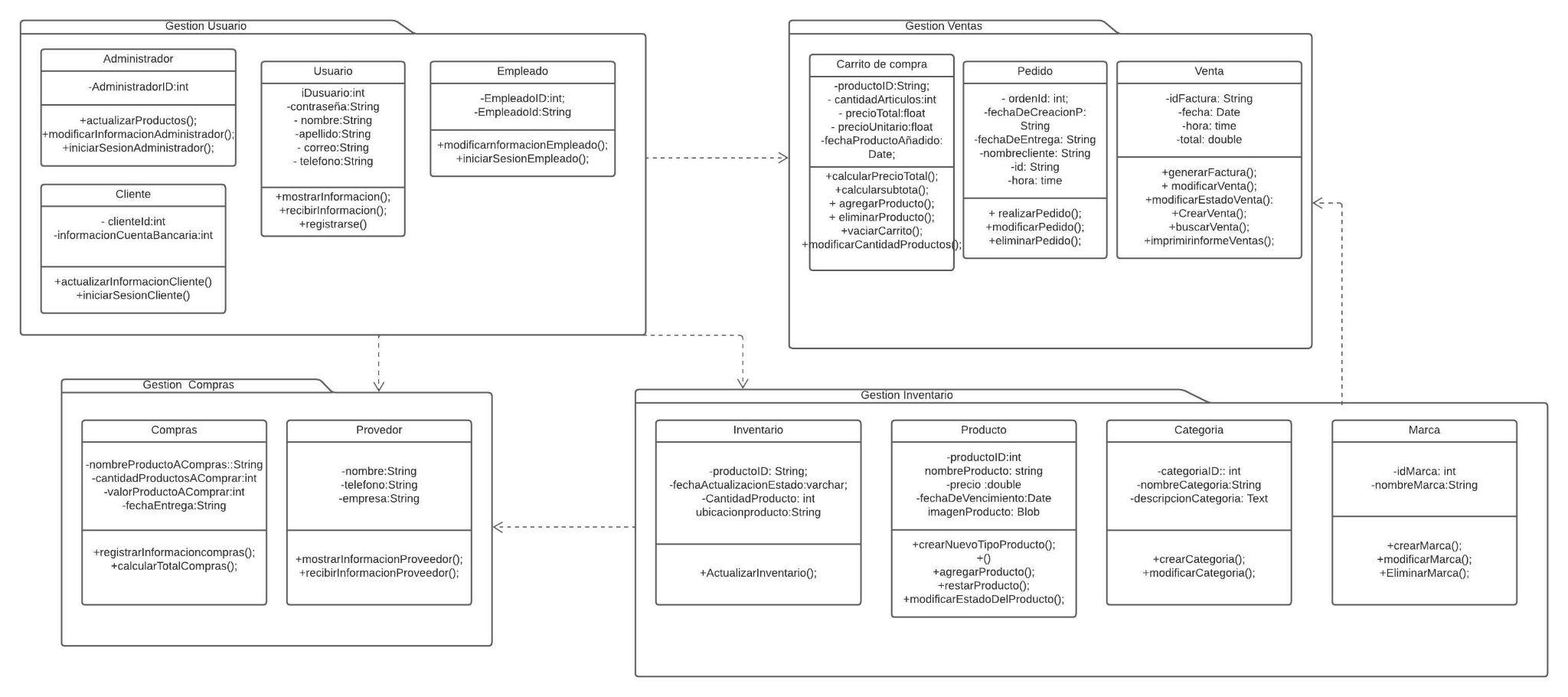
****

## VISTA DE IMPLEMENTACIÓN

### Diagrama de Componentes

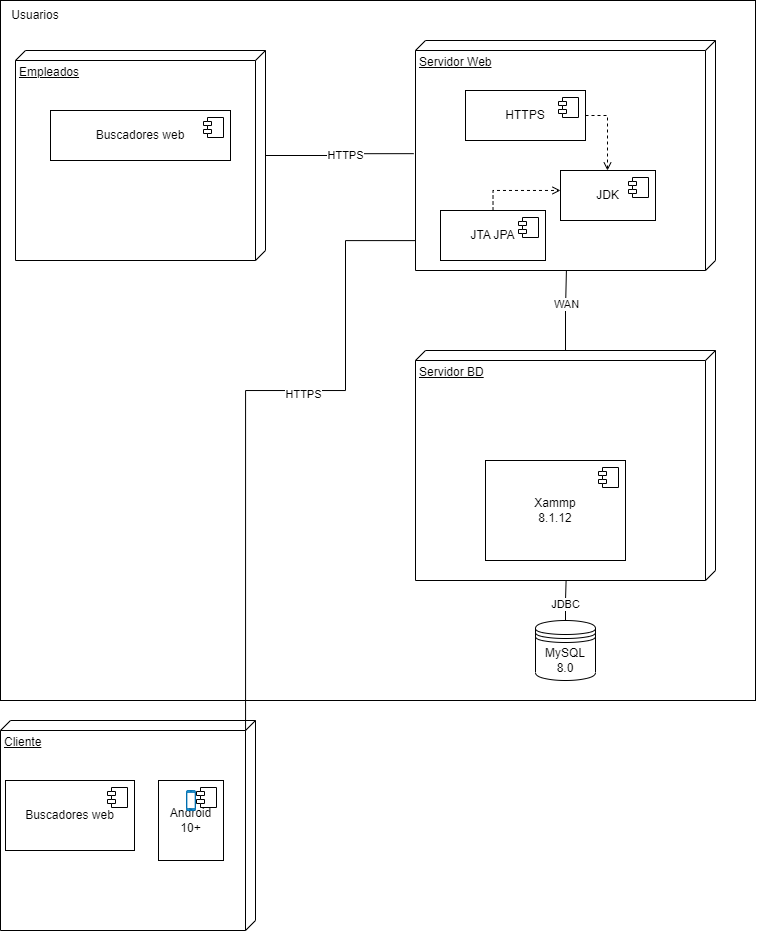


### Diagrama de Paquetes

****

## VISTA DE DESPLIEGUE

### Diagrama de despliegue



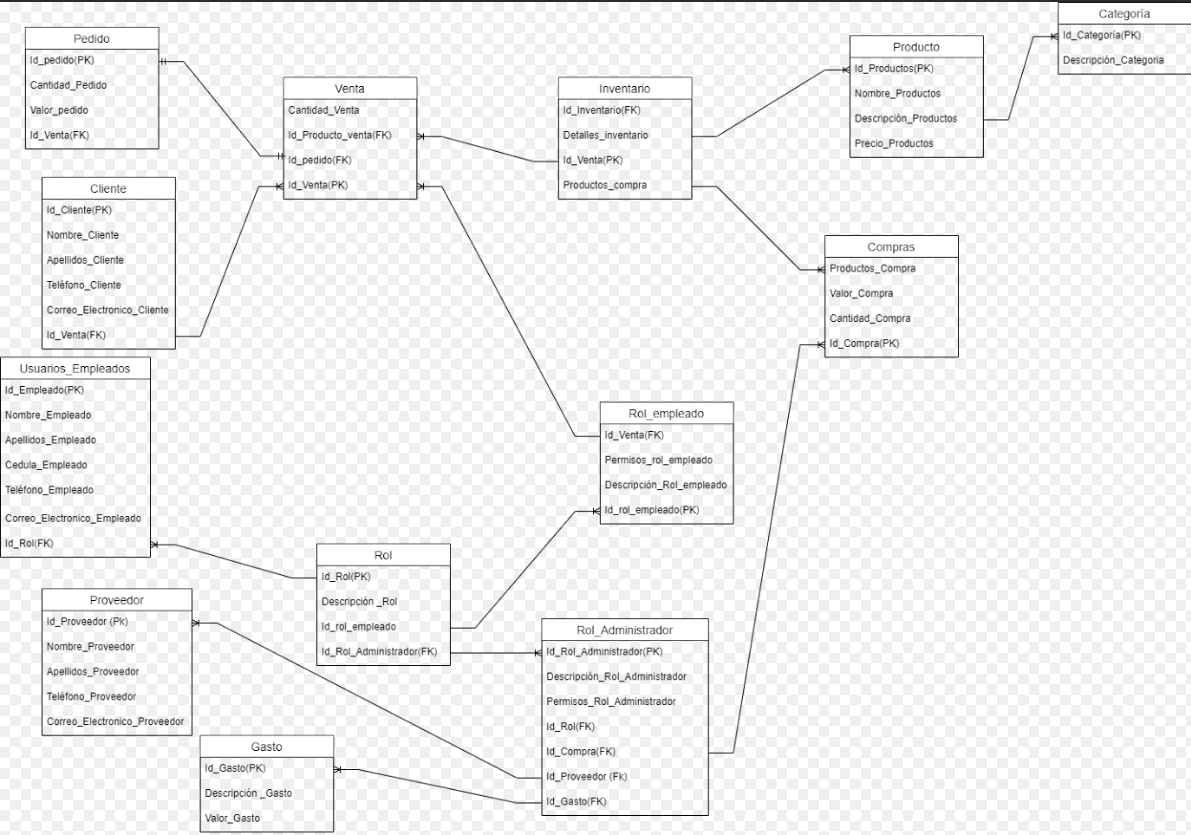
# Arquitectura en capas

* Cuadro de Tecnologías por capas

| **CAPA** | **TECNOLOGÍAS A USAR** |
| --- | --- |
| Presentación | * HTML * CSS * BOOTSTRAP * Javascript |
| Lógica | * JAVA |
| Persistencia | * DAO * MySQL * JAVA API |

# VISTA DE DATOS

## Modelo Relacional normalizado tercera forma normal de la base de datos (SGBD).



# Definición de Interfaces de Usuario

* Página principal

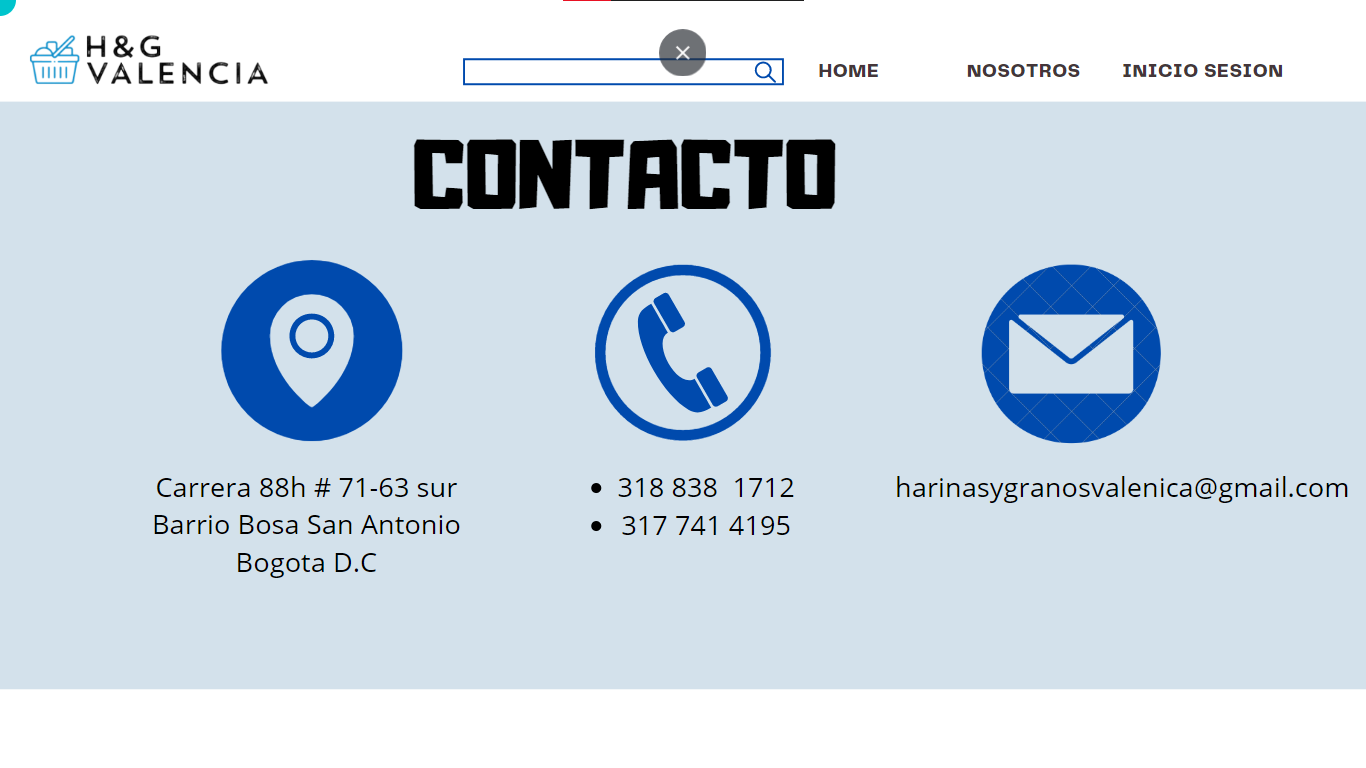


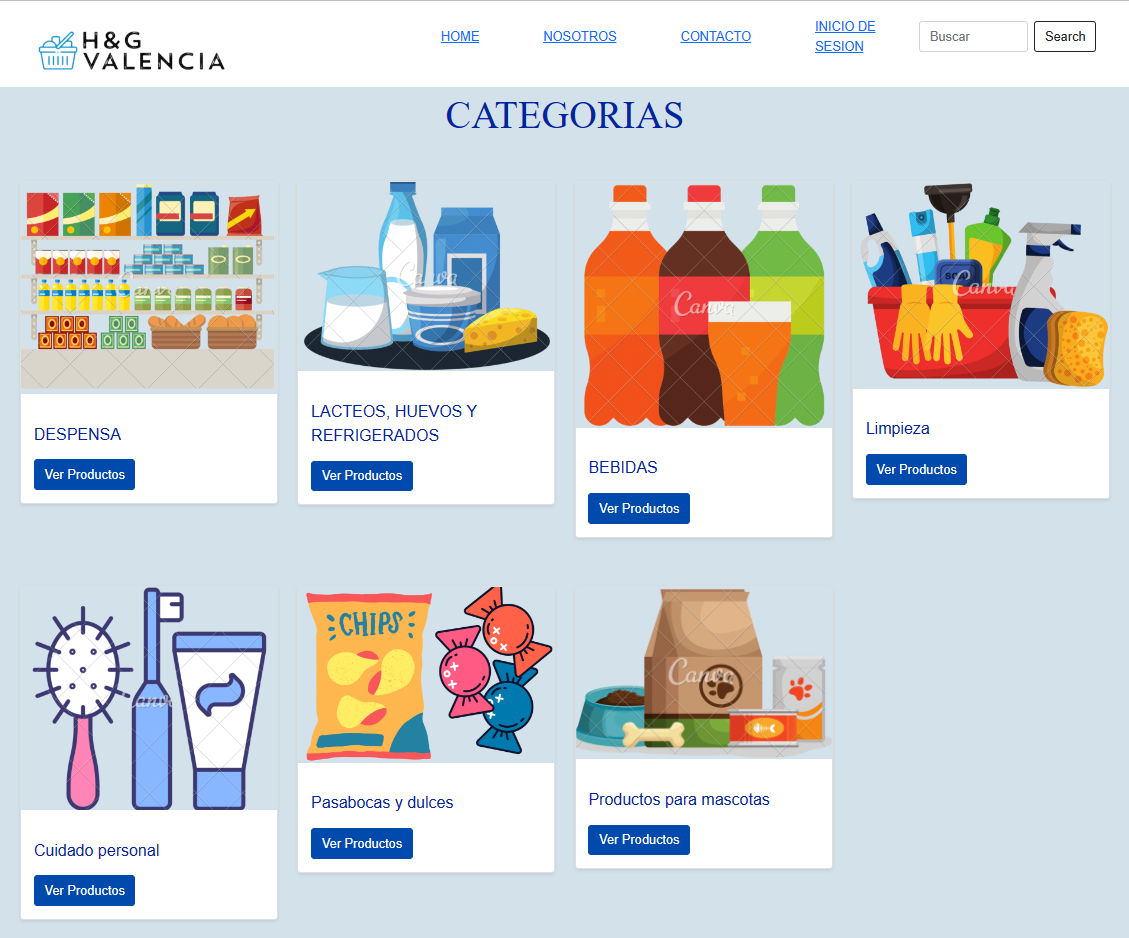
Formularios



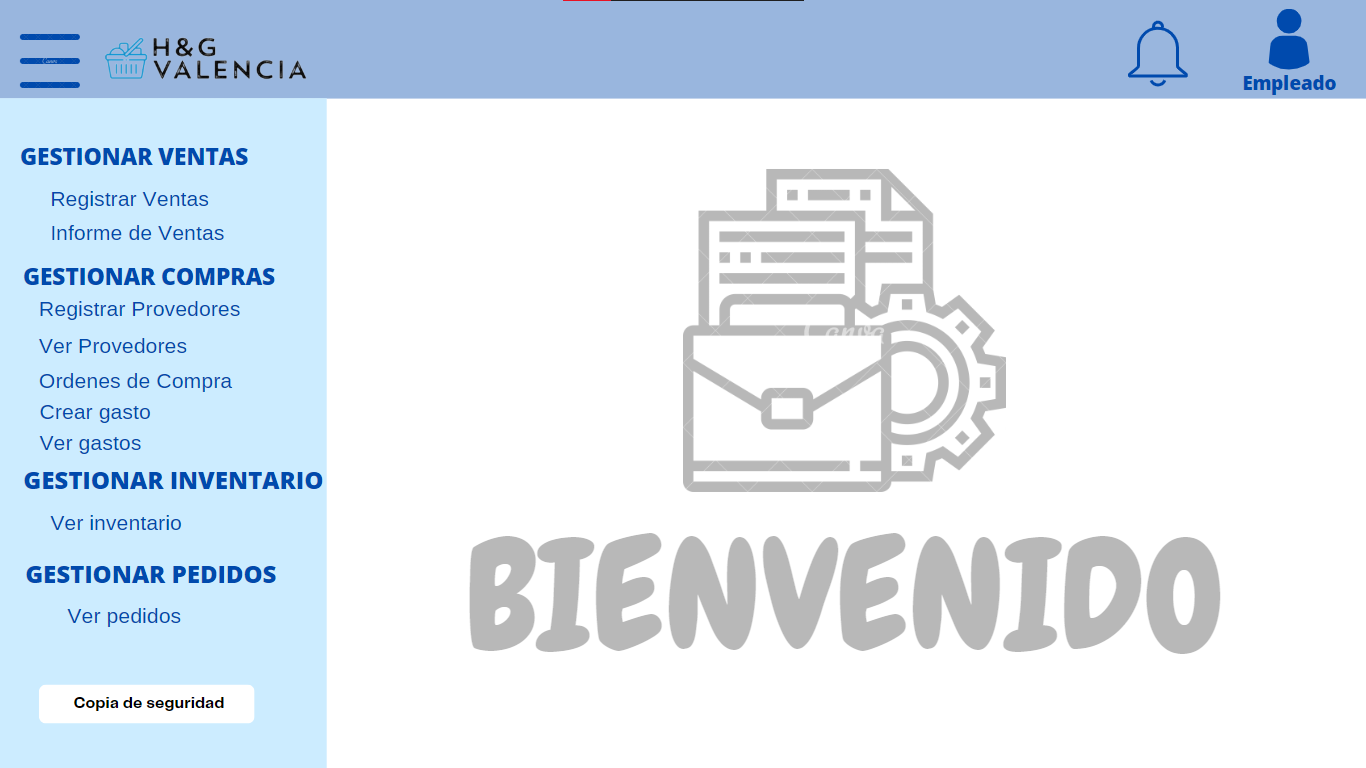












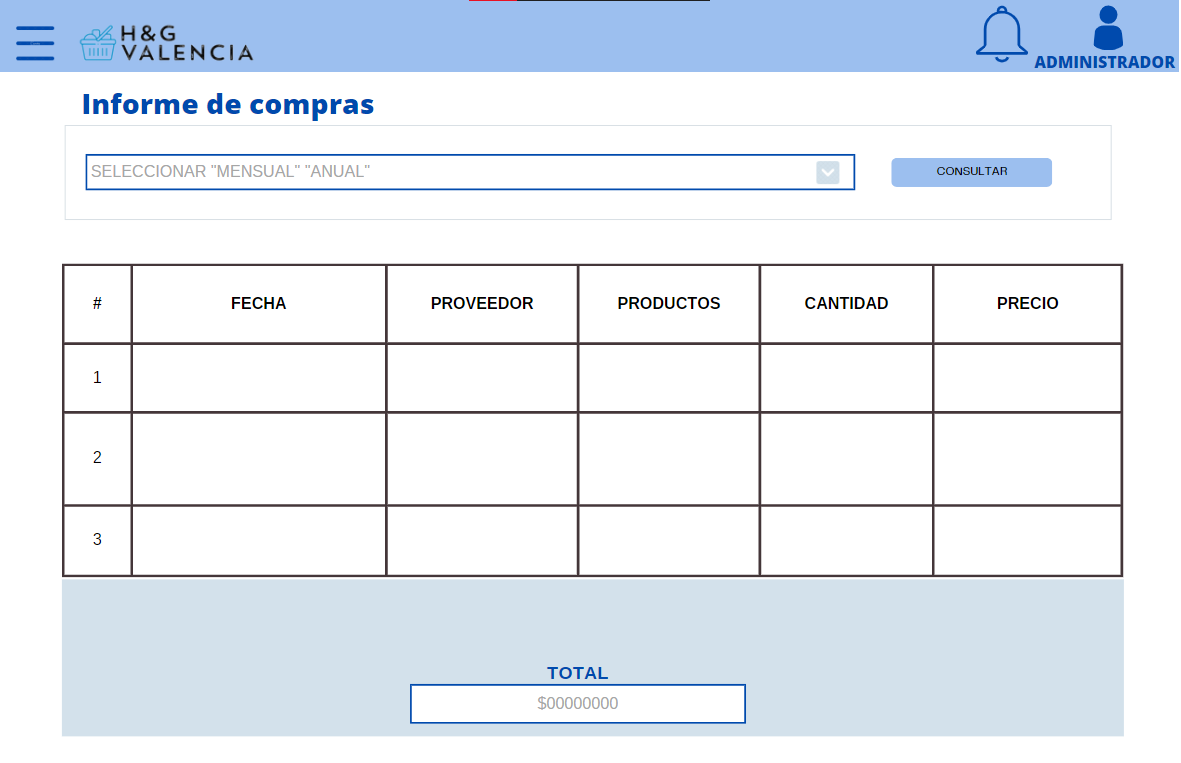


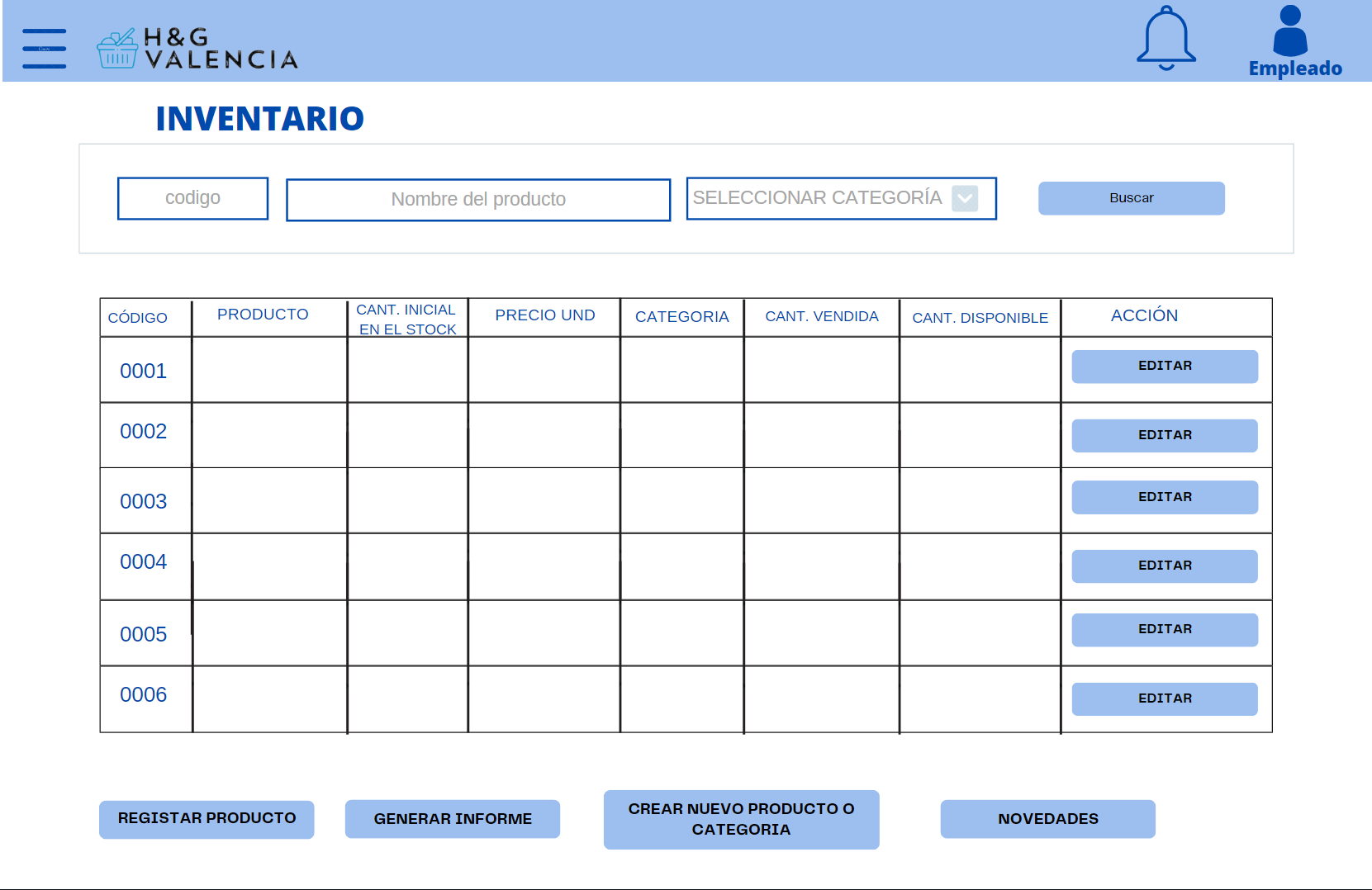
















Errores





# Características Generales de Calidad

## Tamaño y performance x

* Tiempo de respuesta en el acceso a la Base de Datos:
* Tiempo de respuesta de transacciones:
* Espacio en disco para el cliente:
* Espacio en disco para el servidor de Base de datos:

Puntos 7.2. al 7.7., se debe argumentar por cada atributo de calidad, como se cumple el mismo para el Software.

## Calidad

El sistema de información cumplirá con los requisitos de calidad necesarios para satisfacer las necesidades del usuario final. El sistema de información estará diseñado para recopilar, almacenar, procesar y presentar datos de manera precisa, oportuna, relevante y confiable para ayudar a mejorar el manejo de la información de los procesos internos de la microempresa.

## Usabilidad

El sistema de información cumplirá con los requisitos de usabilidad, ya que el sistema será de fácil uso, para que así los usuarios puedan realizar lo necesario en el sistema de forma rápida y eficiente.

## Eficiencia

El sistema de información cumplirá con los requisitos de eficiencia necesaria para satisfacer las necesidades de los usuarios finales. El sistema está diseñado para registrar, ver, modificar y manejar la información de los procesos de la tienda, mejorar la gestión de información de la empresa.

## Seguridad

El sistema de información cumplirá con los requisitos de seguridad, ya que contamos con unas buenas prácticas para proteger la información y el sistema en sí. Ya que para prevenir errores en esta área, realizamos pruebas de seguridad desde el inicio, hasta el final del desarrollo del sistema y poder brindar actualizaciones y parches de seguridad si observamos vulnerabilidades, para así mantener el sistema al día y cumplir con los requisitos de seguridad necesarios para proteger la información.

## Confiabilidad

El sistema de información cumplirá con los requisitos de confiabilidad necesarios. El sistema de información será capaz de realizar sus funciones de manera consistente y precisa. Un sistema confiable debe ser capaz de manejar grandes cantidades de datos de ventas, inventario y clientes sin fallas o interrupciones significativas.

## Mantenimiento

Para cumplir con este requerimiento el sistema le realizará un seguimiento constante para detectar cualquier error o falla, con el fin de actuar de manera proactiva. Otro factor importante para realizar un plan de mantenimiento es monitorear el rendimiento del sistema, para detectar cualquier problema o cuello de botella. por último, dependiendo de los factores mencionados antes se creará un plan de mantenimiento para así determinar si se necesitan mantenimientos preventivos o correctivos y la frecuencia con la que se llevarán a cabo.

## Estándares (Normas de calidad): Mencionar dos normas de calidad de Desarrollo de Software y argumentar cómo esas normas aplican para el software.

25010: Con esta ISO vamos a asegurar un marco de calidad en nuestro software, con características como: funcionalidad, rendimiento,seguridad, mantenibilidad, etc..

25012: Al usar esta ISO se va a asegurar la calidad de los datos, también facilitan el entender sobre el grado en el que los datos son requeridos y definidos con su respectiva características: Exactitud, Completitud, Consistencia, Credibilidad, Actualidad, Accesibilidad, etc..