**Proyecto MagikStock**

**(SAD) Software Architecture Document**

**Versión 1.0**

**Integrantes:**

Nicolas Moreno

Cristian Núñez

Catary Rodriguez

**Docente:**

Rocio Contreras

**Asignatura:**

Capstone

**Sección:**

003D

**Identificación de Documento**

| **Identificación** | 0 |
| --- | --- |
| **Proyecto** | MagikStock |
| **Versión** | 1.0 |

| **Documento mantenido por** | Nicolas Moreno, Cristian Núñez y Catary Rodriguez |
| --- | --- |
| **Fecha de última revisión** | 09/09/2024 |
| **Fecha de próxima revisión** | 20/10/2024 |

| **Documento aprobado por** | Catary Rodriguez |
| --- | --- |
| **Fecha de última aprobación** | 30/09/2024 |

**Historia de cambios**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 09/09/2024 | 1.0 | Se comienza el desarrollo del documento. | Catary Rodriguez |

**Tabla de Contenidos**

[**Índice de figuras 2**](#_heading=h.gjdgxs)

[**Introducción 3**](#_heading=h.30j0zll)

[Contexto del Problema 3](#_heading=h.1fob9te)

[**1**](#_heading=h.3znysh7) **Alcance 3**

[**2**](#_heading=h.2et92p0) **Referencias 3**

[**3**](#_heading=h.tyjcwt) **Arquitectura de Software 3**

[**4**](#_heading=h.3dy6vkm) **Objetivos y Restricciones de la Arquitectura 4**

[Objetivos de la arquitectura 4](#_heading=h.1t3h5sf)

[Restricciones de la Arquitectura 4](#_heading=h.4d34og8)

[**5**](#_heading=h.2s8eyo1) **Arquitectura Lógica 5**

[Parte Estructural 5](#_heading=h.17dp8vu)

[Parte Dinámica 5](#_heading=h.3rdcrjn)

[**6**](#_heading=h.26in1rg) **Arquitectura de Procesos 5**

[**7**](#_heading=h.lnxbz9) **Arquitectura de desarrollo 6**

[**8**](#_heading=h.35nkun2) **Arquitectura física 7**

[**9**](#_heading=h.44sinio) **Escenarios 7**

[Modelo de Casos de Uso 7](#_heading=h.2jxsxqh)

[Especificación de Casos de Uso Relevantes 7](#_heading=h.z337ya)

[Especificación de los Escenarios de Calidad Relevantes 8](#_heading=h.3j2qqm3)

[**10**](#_heading=h.1y810tw) **Tamaño y desempeño 9**

**Índice de figuras**

**Introducción**

Magikoffee es una cafetería que recientemente ha expandido su negocio al abrir una nueva sucursal, una heladería, lo que ha generado la necesidad de un sistema que permita gestionar su inventario de manera eficiente y centralizada. Debido a que ambas sucursales comparten un inventario común, la gestión manual se ha vuelto ineficiente y propensa a errores. Por ello, se desarrollará un sistema de control de stock automatizado y centralizado con una interfaz web, que permitirá el monitoreo en tiempo real, la sincronización de inventarios y la generación de reportes para ambas sucursales.

1. **Alcance**

El alcance de este documento incluye la definición de los casos de uso y escenarios de calidad que serán la base para el desarrollo del sistema. El software se enfocará en la gestión del inventario compartido entre la cafetería y la heladería, proporcionando una solución integral para la administración del stock, la sincronización entre sucursales y la generación de alertas y reportes automáticos.

El proyecto está dirigido a los administradores de Magikoffee, quienes podrán gestionar el inventario de manera eficiente, y a los empleados, quienes interactuarán con el sistema para registrar ventas y recepciones de stock.

1. **Referencias**

A continuación, se listan las referencias utilizadas para el desarrollo de este documento y del sistema:

* ISO/IEC 25010:2011 – Systems and software quality requirements and evaluation (SQuaRE) – System and software quality models.
* Metodología PMBOK 6ª edición
* Documentación del sistema de control de inventario de Magikoffee

**Arquitectura de Software**

La arquitectura del sistema de control de stock de Magikoffee está diseñada siguiendo el enfoque del framework 4+1, con un modelo de capas que facilita la organización y gestión del sistema. Las vistas incluidas en esta versión del documento son:

La arquitectura que ocuparemos será el modelo de capas

* **Vista de Casos de Uso y Escenarios de Calidad**: Describe los casos de uso más significativos, presenta los actores y una descripción de sus casos de uso asociados. De igual forma describe los escenarios de calidad más relevantes para la arquitectura.

**Modelo de Capas:**

* El sistema de control de stock de Magikoffee está diseñado siguiendo un modelo de capas que incluye las siguientes capas principales:
* **Capa de Presentación**: Proporciona la interfaz de usuario basada en web, permitiendo la interacción con el sistema por parte de los administradores y empleados.
* **Capa de Lógica de Negocio**: Contiene la lógica de negocio necesaria para gestionar el inventario, la sincronización de sucursales y la generación de reportes y alertas.
* **Capa de Datos**: Gestiona el acceso y la persistencia de los datos del inventario, incluyendo el almacenamiento de productos, proveedores, ventas y movimientos de stock.

| **Casos de uso** | | | | |
| --- | --- | --- | --- | --- |
| **Código** | **Nombre** | **Actores** | **Descripción** | **Prioridad** |
| CU-001-001 | Monitorear Stock en Tiempo Real | Administradores, Empleados | Permite a los usuarios monitorear el inventario en tiempo real para ambas sucursales. | Alta |
| CU-002-001 | Gestionar Productos | Administradores | Permite agregar, modificar y eliminar productos en el sistema de inventario. | Alta |
| CU-003-001 | Generar Reportes de Inventario | Administradores | Genera reportes diarios o semanales sobre el estado del inventario y ventas. | Media |
| CU-004-001 | Sincronizar Inventario | Sistema | Sincroniza el inventario de ambas sucursales en un sistema centralizado. | Alta |
| CU-005-001 | Configurar Alertas de Stock Bajo | Administradores | Configura alertas cuando el inventario de un producto se encuentre bajo un umbral predefinido. | Alta |
| CU-006-001 | Registrar Entrada de Stock | Empleados, Administradores | Permite registrar la entrada de nuevos productos al inventario al recibir mercancía. | Media |
| CU-007-001 | Administrar Proveedores | Administradores | Permite el registro y la gestión de proveedores asociados al inventario. | Media |
| CU-008-001 | Gestionar Accesos y Roles | Administradores | Gestiona los permisos y roles de acceso de los usuarios del sistema. | Alta |
| CU-009-001 | Acceso Remoto al Sistema | Administradores | Permite a los administradores acceder al sistema desde cualquier dispositivo con conexión a internet. | Alta |
| CU-0010-001 | Realizar Copia de Seguridad | Sistema | Realiza copias de seguridad automáticas del sistema de inventario para prevenir la pérdida de datos. | Media |

| **Escenarios de Calidad** | | | |
| --- | --- | --- | --- |
| **Número** | **Nombre** | **Atributo de calidad** | **Descripción** |
| 1 | Interfaz Web Operativa | Usabilidad | El sistema debe proporcionar una interfaz web que sea operativa y fácil de usar para que los empleados y administradores gestionen el inventario eficientemente. Este requisito es observable y se asocia con la operabilidad. |
| 2 | Facilidad de Uso de la Interfaz | Usabilidad | La interfaz debe ser intuitiva y fácil de entender, permitiendo a los empleados y administradores usar el sistema con una mínima curva de aprendizaje. Esto es observable y está asociado a la comprensibilidad y aprendizaje. |
| 3 | Accesibilidad Remota | Portabilidad | El sistema debe ser accesible desde cualquier dispositivo con conexión a internet, adaptándose a diferentes plataformas y resoluciones. Es un atributo observable asociado con la capacidad de ser instalada y la adaptabilidad. |
| 4 | Seguridad de Acceso | Seguridad | El sistema debe asegurar la confidencialidad y autenticidad, protegiendo la información sensible del inventario mediante control de accesos basado en roles. Es un requisito no observable que garantiza la seguridad del sistema. |
| 5 | scalabilidad del Sistema | Mantenibilidad | El sistema debe ser modular y capaz de expandirse para adaptarse al crecimiento del negocio sin requerir grandes modificaciones. Es un atributo no observable asociado con la modularidad y capacidad de expansión. |
| 6 | Backup y Recuperación Automática | Fiabilidad | El sistema debe realizar copias de seguridad automáticas periódicamente y permitir la recuperación de datos en caso de falla, asegurando la continuidad de las operaciones. Es un atributo no observable relacionado con la capacidad de recuperación. |

* **Vista de Metas y Restricciones**: Describe restricciones tecnológicas, normativas, estándares, etc., los cuales influyen sobre las decisiones arquitectónicas, del producto y del proceso de desarrollo.

**Metas del Proyecto**

1. **Optimización del Control de Inventario:**

* Automatizar la gestión de inventario para minimizar errores humanos y mejorar la eficiencia operativa de ambas sucursales.

1. **Monitoreo en Tiempo Real:**

* Proveer a los administradores y empleados la capacidad de monitorear el inventario en tiempo real para tomar decisiones informadas y oportunas sobre la reposición de stock.

1. **Facilitar la Toma de Decisiones:**

* Generar reportes automáticos diarios o semanales que proporcionen datos relevantes para la planificación y la toma de decisiones estratégicas.

1. **Sincronización Centralizada:**

* Asegurar que la información del inventario esté sincronizada entre la cafetería y la heladería, evitando duplicidades y desabastecimiento.

1. **Seguridad y Acceso Controlado:**

* Implementar un sistema de control de acceso basado en roles para proteger la información del inventario y garantizar que solo usuarios autorizados puedan realizar modificaciones.

1. **Escalabilidad y Adaptabilidad:**

* Diseñar un sistema escalable que permita integrar nuevas sucursales o funcionalidades a medida que el negocio crezca.

1. **Accesibilidad Remota:**

* Asegurar que el sistema sea accesible desde cualquier dispositivo con conexión a internet para permitir a los administradores monitorear y gestionar el inventario desde cualquier lugar.

1. **Facilidad de Uso:**

* Desarrollar una interfaz web intuitiva y fácil de usar para minimizar la curva de aprendizaje y los errores de los usuarios.

1. **Restricciones del Proyecto**
2. **Presupuesto Limitado**:
   * El desarrollo y la implementación del sistema deben mantenerse dentro del presupuesto asignado por Magikoffee.
3. **Tiempo de Implementación**:
   * El sistema debe estar completamente implementado y funcional en un plazo máximo de 6 meses para cumplir con los objetivos del negocio.
4. **Compatibilidad con Dispositivos Existentes**:
   * El sistema debe ser compatible con los dispositivos y sistemas operativos que actualmente utilizan ambas sucursales (PC, tablets, smartphones).
5. **Capacidad de Infraestructura**:
   * El sistema debe operar dentro de las limitaciones de la infraestructura tecnológica actual de Magikoffee, como el ancho de banda y la capacidad de los servidores.
6. **Normativas de Seguridad**:
   * El sistema debe cumplir con las normativas de seguridad de datos y protección de la información, garantizando la confidencialidad y privacidad de la información de inventario.
7. **Acceso a Internet**:
   * Dado que el sistema es web, depende de una conexión a internet estable para su funcionamiento. Las sucursales deben contar con acceso constante a internet para asegurar la operatividad del sistema.
8. **Almacenamiento y Backup de Datos**:
   * El sistema debe implementar un mecanismo de copias de seguridad automáticas que se ajuste a la capacidad de almacenamiento disponible en la infraestructura actual.
9. **Disponibilidad y Fiabilidad**:
   * El sistema debe ser fiable y estar disponible al menos un 99% del tiempo, ya que se utilizará para operaciones críticas en ambas sucursales.
10. **Uso de Tecnologías Estándar**:
    * Deben usarse tecnologías que sean estándar y ampliamente aceptadas para asegurar la sostenibilidad y la facilidad de mantenimiento del sistema.

* **Vista Lógica**: Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias. Usa el view type Módulos para representar la estructura lógica y el view type Componentes y Conectores para representar el comportamiento.

Diagrama

Descripción generada automáticamente

* **Vista de Procesos**: Describe los procesos involucrados para darle sentido a la ejecución del sistema, así como sus relaciones de comunicación y sincronización.
* **Vista de Implementación**: Describe los componentes de deployment construidos y sus dependencias.

1. **Objetivos y Restricciones de la Arquitectura**

A continuación se revisan las metas y restricciones de la arquitectura.

**Objetivos de la arquitectura**

De acuerdo con las reuniones y al análisis de los requerimientos, se listan los principales conductores iniciales de la arquitectura los cuales corresponden a las metas arquitectónicas iniciales:

* **Usabilidad:** Operabilidad, Comprensibilidad, Aprendizaje
* **Portabilidad**: Adaptabilidad.
* **Seguridad**: Autenticidad, Confidencialidad.
* **Compatibilidad**:Coexistencia.
* **Mantenibilidad:** Modularidad, Capacidad de expansión.
* **Fiabilidad:** Capacidad de recuperación

**Restricciones de la Arquitectura**

Las restricciones de la arquitectura de software planteada para el sistema contempla aspectos importantes en las etapas del desarrollo, a continuación se describe las siguientes restricciones del diseño de la arquitectura:

* El proyecto se realizará sobre una arquitectura Web
* La construcción de la interfaz Web tendrá un plazo máximo de 12 semanas.
* El software tendrá la facilidad de poder visualizarse mediante una conexión a Internet por medio de un navegador.
* La implementación de la base de datos tendrá un plazo máximo de 2 semanas.
* El sistema operativo de desarrollo es Microsoft Windows.
* El protocolo de comunicación de la capa de aplicaciones y la base de datos será TCP/IP.

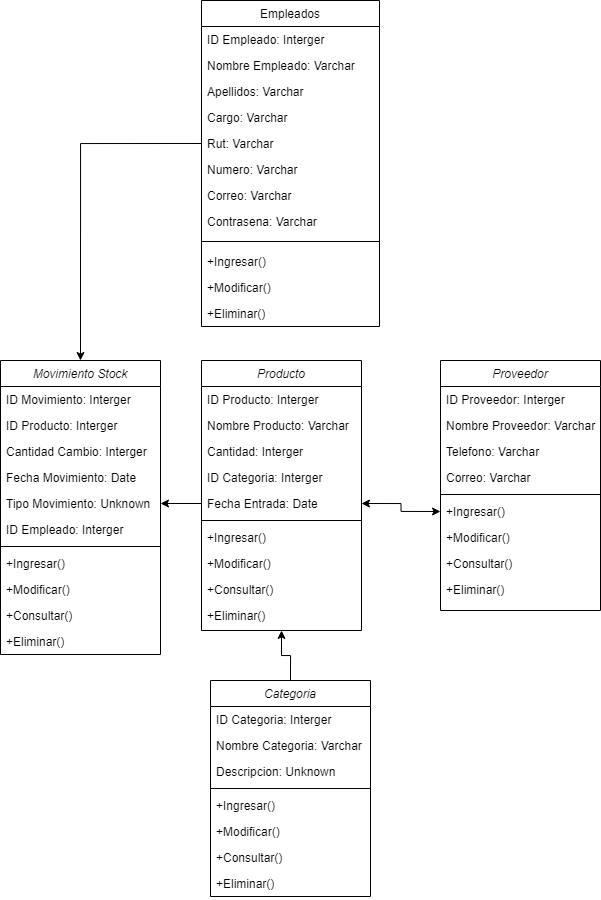
1. **Arquitectura Lógica**

A continuación se presenta una vista lógica de la aplicación expresado en dos diagramas, uno de ellos que muestra la parte estructural o estática de la aplicación (módulos), y otra vista que representa la parte dinámica (componentes y conectores).

**Parte Estructural**

En el siguiente diagrama de clases se observa que el principal módulo….

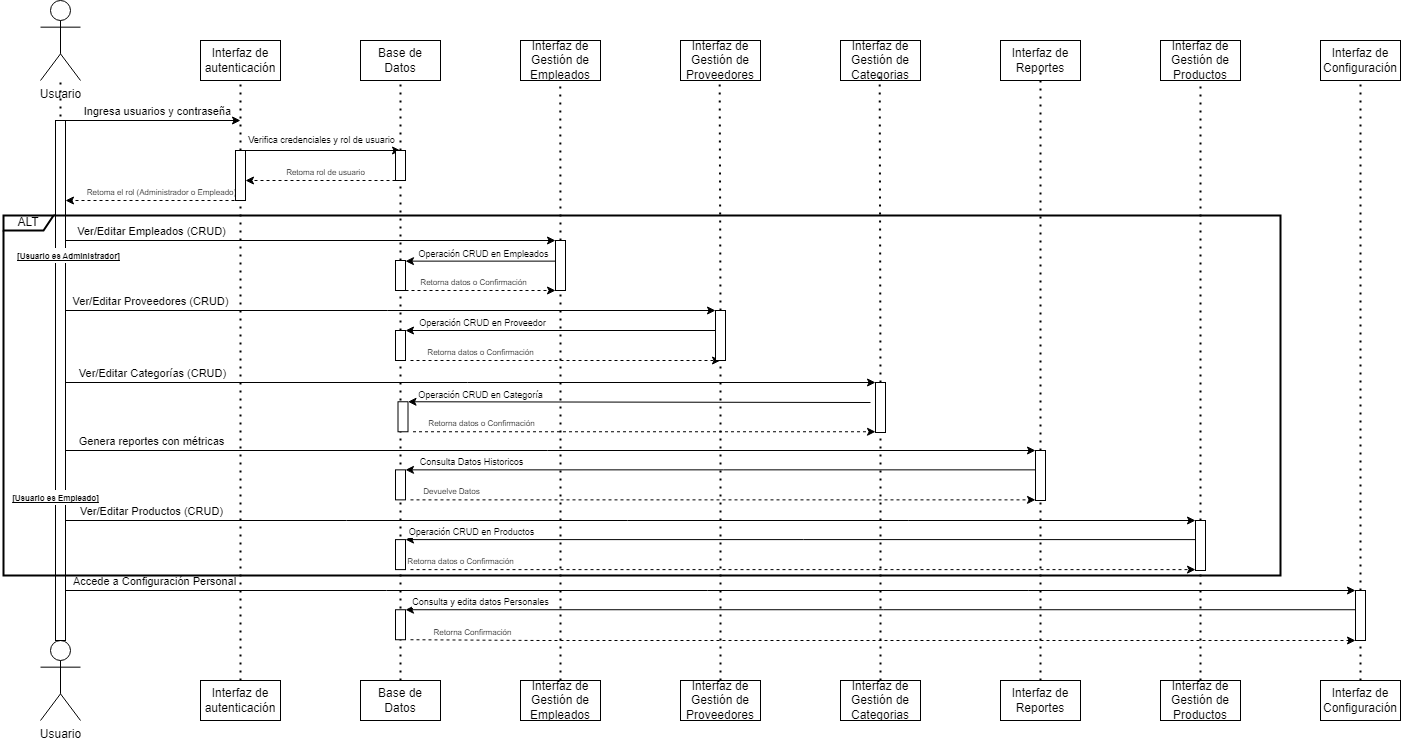
**Ilustración 1: Diagrama de clases**



**Parte Dinámica**

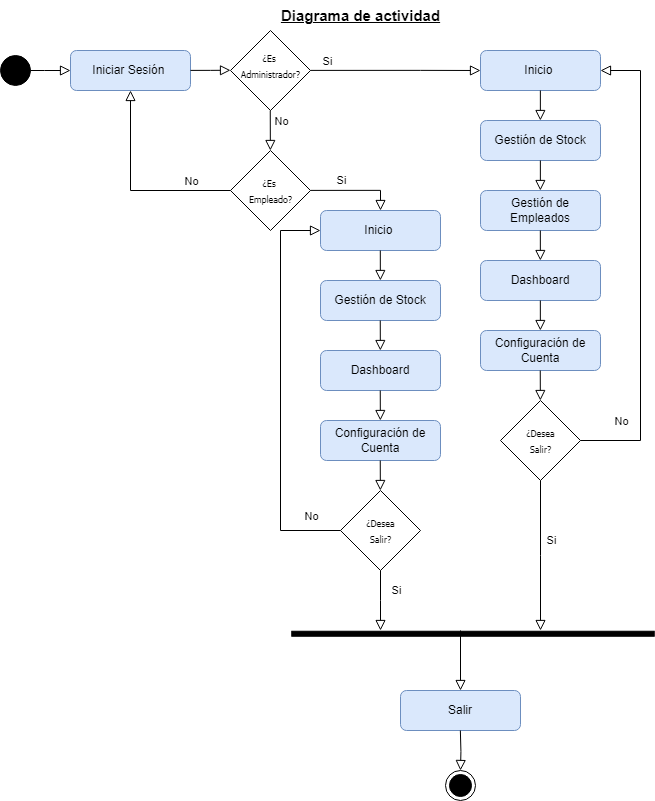
La parte dinámica….

**Ilustración 2: Diagrama de secuencia**

****

1. **Arquitectura de Procesos**

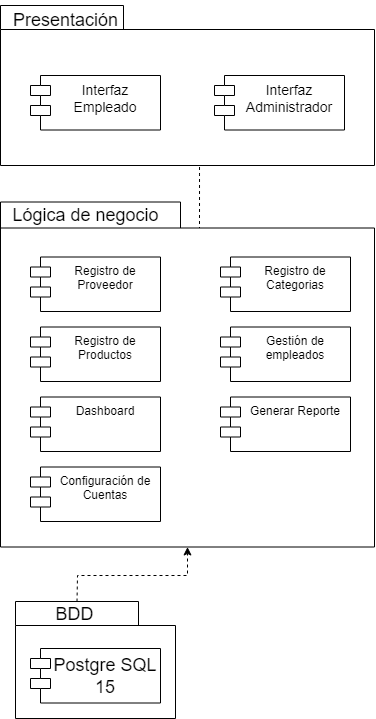
A continuación se muestra una vista de procesos, en la cual se observa que:

**Ilustración 3: Diagrama de actividad**

1. **Arquitectura de desarrollo**

En esta vista se aprecia que existirán dos módulos principales que contendrán distintas funcionalidades de la aplicación. A continuación se describen:

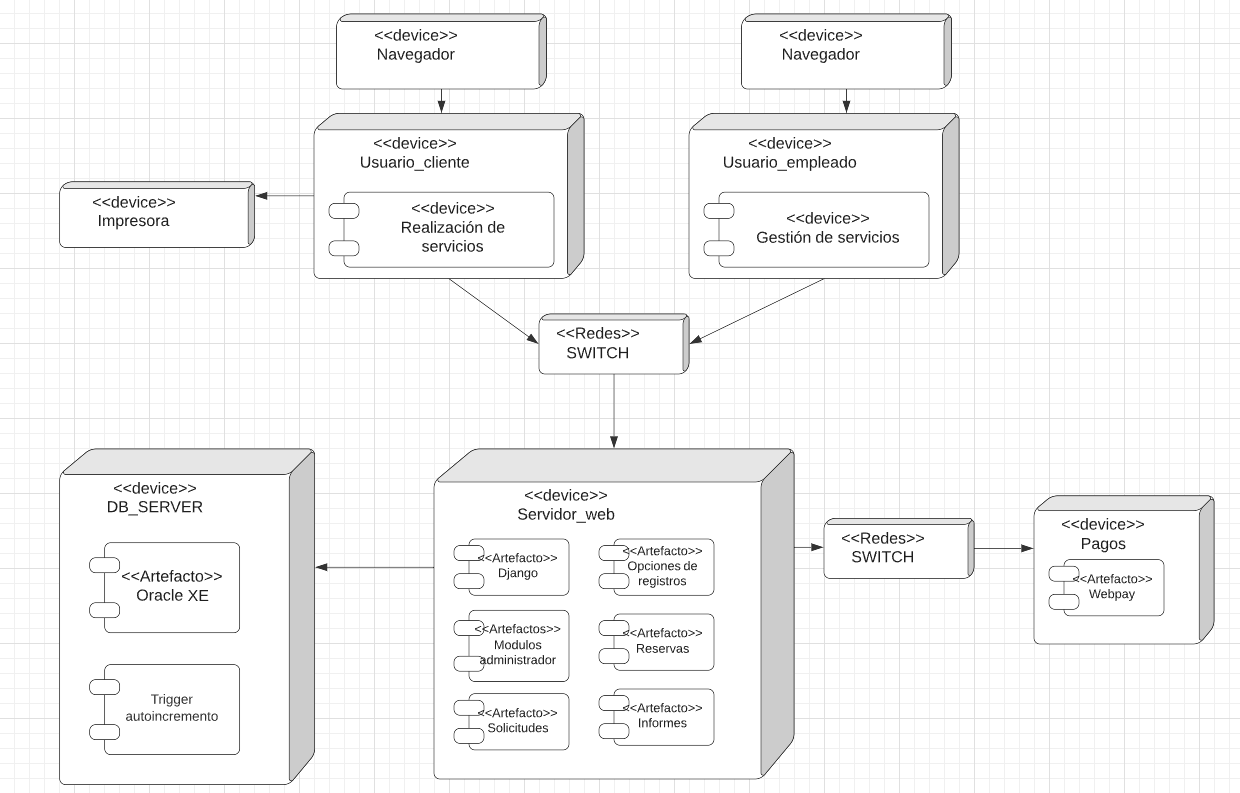
**Ilustración 4: Vista de Implementación (Diagramas de componentes o paquetes)**

****

1. **Arquitectura física**

En esta vista se despliegan los nodos que participan con el sistema. Los nodos principales son los nodos Servidor de Integración. Características a continuación:

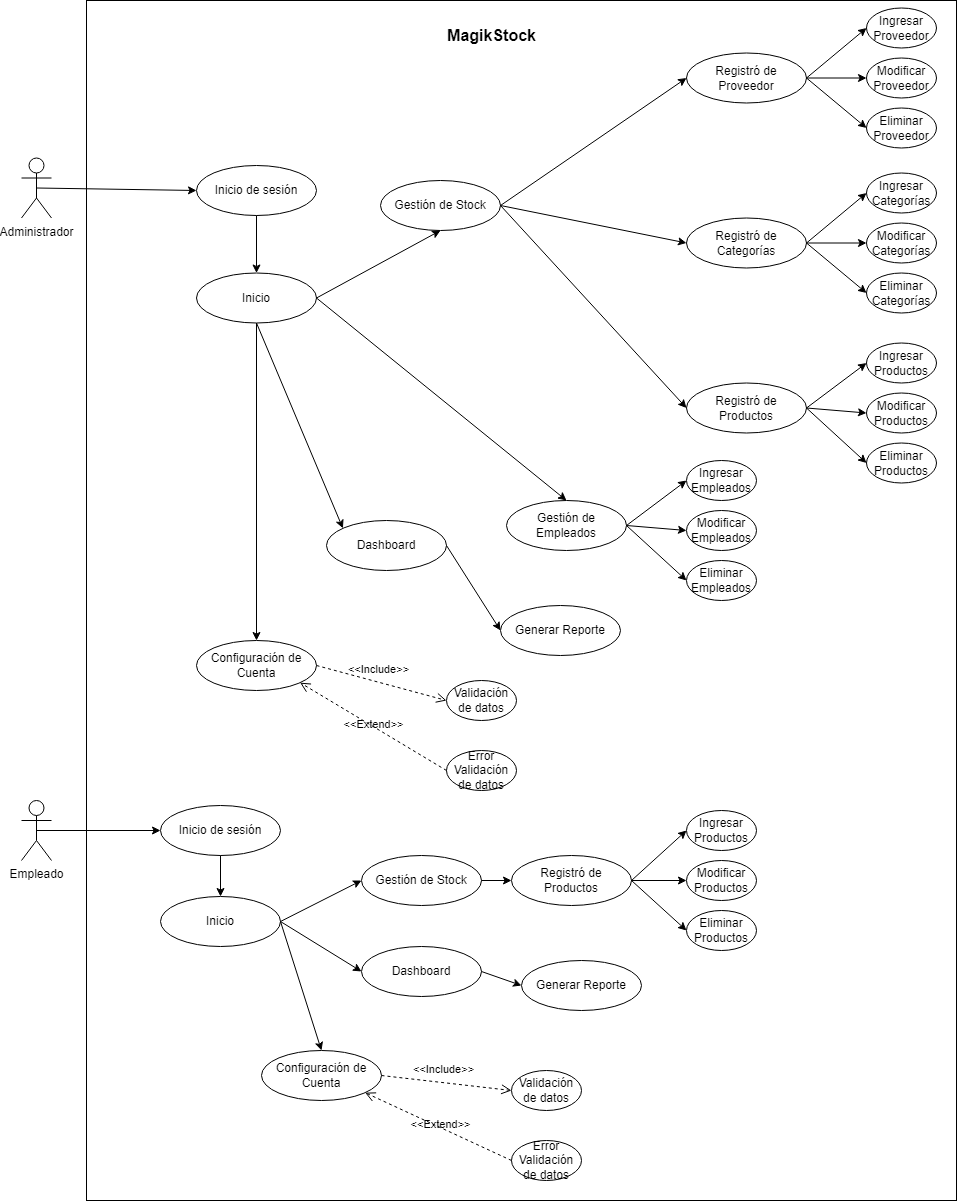
**Ilustración 6: Diagrama de Despliegue**

****

1. **Escenarios**

Esta sección describe en detalle el conjunto de escenarios funcionales y no funcionales que obtuvieron la mayor prioridad en el análisis. Para esto se presenta y describe el diagrama de casos de uso y los casos de uso prioritarios, así como los escenarios en que uno o más atributos de calidad se ven involucrados de manera significativa.

El modelo de casos de uso puede ser encontrado en el documento “Casos de Uso”.



**Especificación de Casos de Uso Relevantes**

Los casos de uso considerados los más relevantes para el desarrollo de la arquitectura fueron determinados. Los criterios usados para dicha determinación fueron:

* Su implementación implica varios nodos de la vista de despliegue.
* Su implementación es de alto riesgo.
* Incluye muchos conceptos y relaciones del dominio.
* Incluye posibles escenarios críticos de calidad.

A continuación se listan los casos de uso relevantes, los cuales pueden ser encontrados con su especificación detallada en el documento “Casos de Uso”.

| **Casos de uso** | | | |
| --- | --- | --- | --- |
| **Código** | **Nombre** | **Actores** | **Prioridad** |
| CU-001-001 | Iniciar sesión | Administrador, Empleado | Muy alta |
| CU-002-001 | Gestión de Stock | Administrador, Empleado | Alta |
| CU-002-002 | Registro de proveedor | Administrador | Alta |
| CU-002-003 | Registro de Categorías | Administrador | Alta |
| CU-002-004 | Registro de Productos | Administrador, Empleado | Alta |
| CU-003-001 | Gestión de Empleados | Administrador | Muy Alta |
| CU-004-001 | Dashboard | Administrador, Empleado | Alta |
| CU-002-007 | Configuración de cuenta | Administrador, Empleado | Alta |

**Especificación de los Escenarios de Calidad Relevantes**

Después de un análisis en conjunto con los stakeholders, los escenarios de calidad se expresan a continuación:

**Se agregaron los antiguos escenarios de calidad con sus respectivas correcciones y también se añadieron los nuevos.**

| Escenario de Calidad N°1 | |
| --- | --- |
| Atributo de Calidad Asociado (Característica): Usabilidad  Sub Característica: Operabilidad | |
| Descripción: Un sistema de software cuya interfaz sea web, accesible para los empleados y administradores, permitiendo el manejo eficiente del stock. | |
| **Diagrama  Descripción generada automáticamente** | |
| Fuente del Estímulo: | Usuario |
| Estímulo: | El usuario quiere acceder al sistema desde un navegador web. |
| Artefacto: | Sistema |
| Ambiente: | Producción |
| Respuesta: | El sistema debe permitir la gestión completa del inventario desde una interfaz web. |
| Medida de Respuesta: | El usuario podrá interactuar con el sistema en un entorno web, con acceso completo a las funcionalidades de gestión de inventario. |

| Escenario de Calidad N°2 | |
| --- | --- |
| Atributo de Calidad Asociado (Característica): Usabilidad  Sub Característica: Compresibilidad, Aprendizaje | |
| Descripción: La interfaz debe ser intuitiva y fácil de usar para reducir errores humanos en el manejo del stock. | |
| **Diagrama  Descripción generada automáticamente** | |
| Fuente del Estímulo: | Usuario (Administrador/Empleado) |
| Estímulo: | El usuario realiza tareas de gestión de inventario y necesita entender la interfaz sin capacitación extensa. |
| Artefacto: | Interfaz de usuario |
| Ambiente: | Producción |
| Respuesta: | El sistema proporciona una interfaz clara y fácil de aprender, reduciendo el tiempo de capacitación y errores operativos. |
| Medida de Respuesta: | El usuario es capaz de utilizar las funciones clave sin asistencia en un tiempo reducido, y con una tasa de errores mínima en las primeras interacciones. |

| Escenario de Calidad N°3 | |
| --- | --- |
| Atributo de Calidad Asociado (Característica): Portabilidad  Sub Característica: Capacidad de ser instalada, Adaptabilidad | |
| Descripción: El sistema debe ser accesible desde cualquier dispositivo con conexión a internet para monitoreo remoto. | |
| **Diagrama  Descripción generada automáticamente** | |
| Fuente del Estímulo: | Administrador |
| Estímulo: | El administrador desea acceder al sistema desde un dispositivo móvil o fuera de la red interna. |
| Artefacto: | Sistema web |
| Ambiente: | Producción |
| Respuesta: | El sistema permite el acceso remoto, adaptándose a múltiples dispositivos y resoluciones de pantalla. |
| Medida de Respuesta: | El sistema es completamente funcional en al menos 95% de los dispositivos móviles y navegadores más comunes, sin pérdida de funcionalidad. |

| Escenario de Calidad N°4 | |
| --- | --- |
| Atributo de Calidad Asociado (Característica): Seguridad  Sub Característica: Confidencialidad, Autenticidad | |
| Descripción: El sistema debe contar con control de accesos basado en roles para proteger la información sensible del inventario. | |
| **Diagrama  Descripción generada automáticamente** | |
| Fuente del Estímulo: | Administrador |
| Estímulo: | Un usuario intenta acceder a datos sensibles sin los permisos correspondientes. |
| Artefacto: | Sistema de control de acceso |
| Ambiente: | Producción |
| Respuesta: | El sistema deniega el acceso a usuarios sin privilegios. |
| Medida de Respuesta: | 100% de los intentos no autorizados son bloqueados, y los intentos se registran para auditoría. |

| **Escenario de Calidad N°5** | |
| --- | --- |
| **Atributo de Calidad Asociado (Característica): Mantenibilidad**  **Sub Característica: Modularidad, Capacidad de expansión** | |
| **Descripción: El sistema debe ser escalable para soportar el crecimiento futuro del negocio (nuevas sucursales).** | |
| **Diagrama  Descripción generada automáticamente** | |
| **Fuente del Estímulo:** | **Desarrollador** |
| **Estímulo:** | **Se planea abrir nuevas sucursales y el sistema debe adaptarse sin grandes modificaciones.** |
| **Artefacto:** | **Arquitectura del sistema** |
| **Ambiente:** | **Desarrollo y producción** |
| **Respuesta:** | **El sistema debe ser ampliado para gestionar el inventario de las nuevas sucursales con mínimo impacto en la arquitectura actual.** |
| **Medida de Respuesta:** | **El sistema puede integrar nuevas sucursales con menos del 10% de esfuerzo en reestructuración o ajustes al código existente.** |

| Escenario de Calidad N°6 | |
| --- | --- |
| Atributo de Calidad Asociado (Característica): Fiabilidad  Sub Característica: Capacidad de recuperación | |
| Descripción: El sistema debe realizar copias de seguridad automáticas periódicamente para prevenir la pérdida de datos. | |
| **Diagrama  Descripción generada automáticamente** | |
| Fuente del Estímulo: | Sistema |
| Estímulo: | El sistema experimenta una pérdida de datos o falla crítica. |
| Artefacto: | Sistema de backup |
| Ambiente: | Producción |
| Respuesta: | El sistema restaura los datos desde la copia de seguridad más reciente sin pérdida significativa de información. |
| Medida de Respuesta: | El tiempo de recuperación de datos es menor a 15 minutos, y la pérdida de datos es inferior a 1 hora de trabajo. |

**Tamaño y desempeño**

Las principales decisiones arquitectónicas se tomaron en consideración de la restricción **Tiempo de Construcción**. Dado que el proyecto debe implementarse en un tiempo ajustado y sin holguras, se privilegió la adopción de una arquitectura conocida y que presente un bajo riesgo en su implementación.

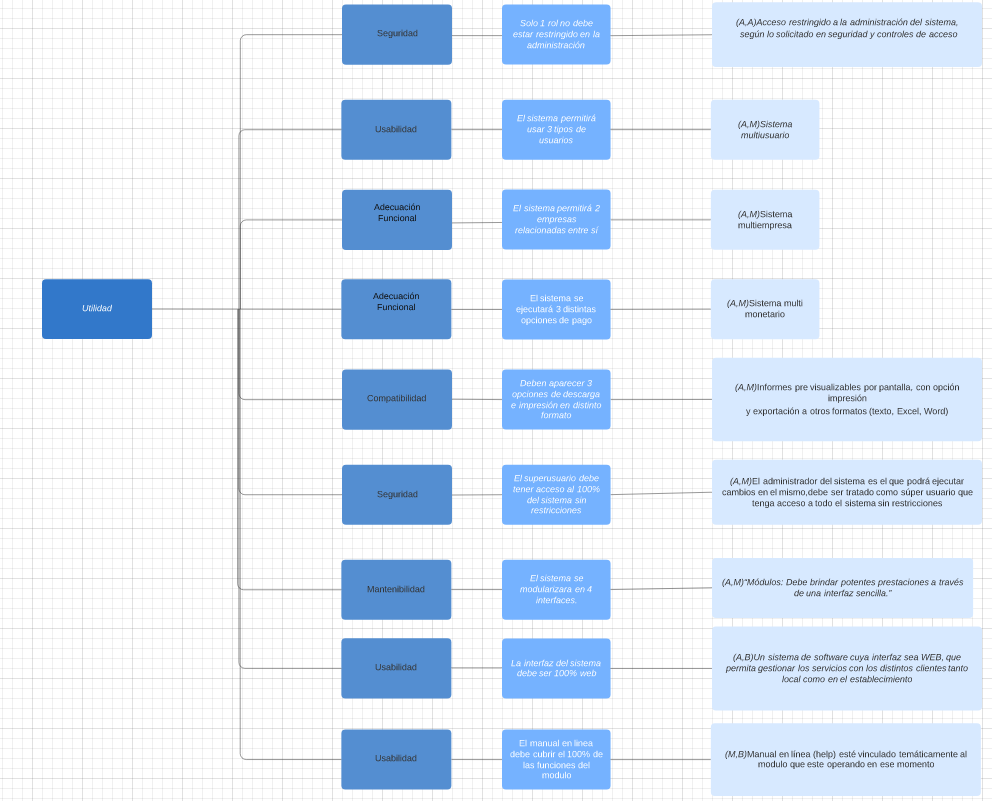
Asimismo, la arquitectura se modularizar con el primer objetivo de separar con cernimientos de forma que permita paralelización en construcción de dichos componentes, y que a su vez sea módulos testeables unitariamente de forma de asegurar (mediante la suite Junit) que cada pieza tenga una baja tasa de fallas.

Un segundo elemento fue considerado en la arquitectura, que corresponde a la restricción de **Infraestructura** con que debe cumplir la aplicación, combinado con el escenario de calidad de **Tolerancia a Fallos**, nos condiciona la modularización de la aplicación en una **aplicación web activa-activa** y una **aplicación de servicios activa-pasiva**.

El escenario de calidad relacionado con la **mantenibilidad** nos conduce al modelamiento pensando en la separación de concernimiento de los componentes y a la utilización del patrón **provider** de forma que el sistema pueda delegar sus requerimientos de información hacia sistemas externos a piezas de software no acopladas que nos permitan su extensibilidad a futuro.

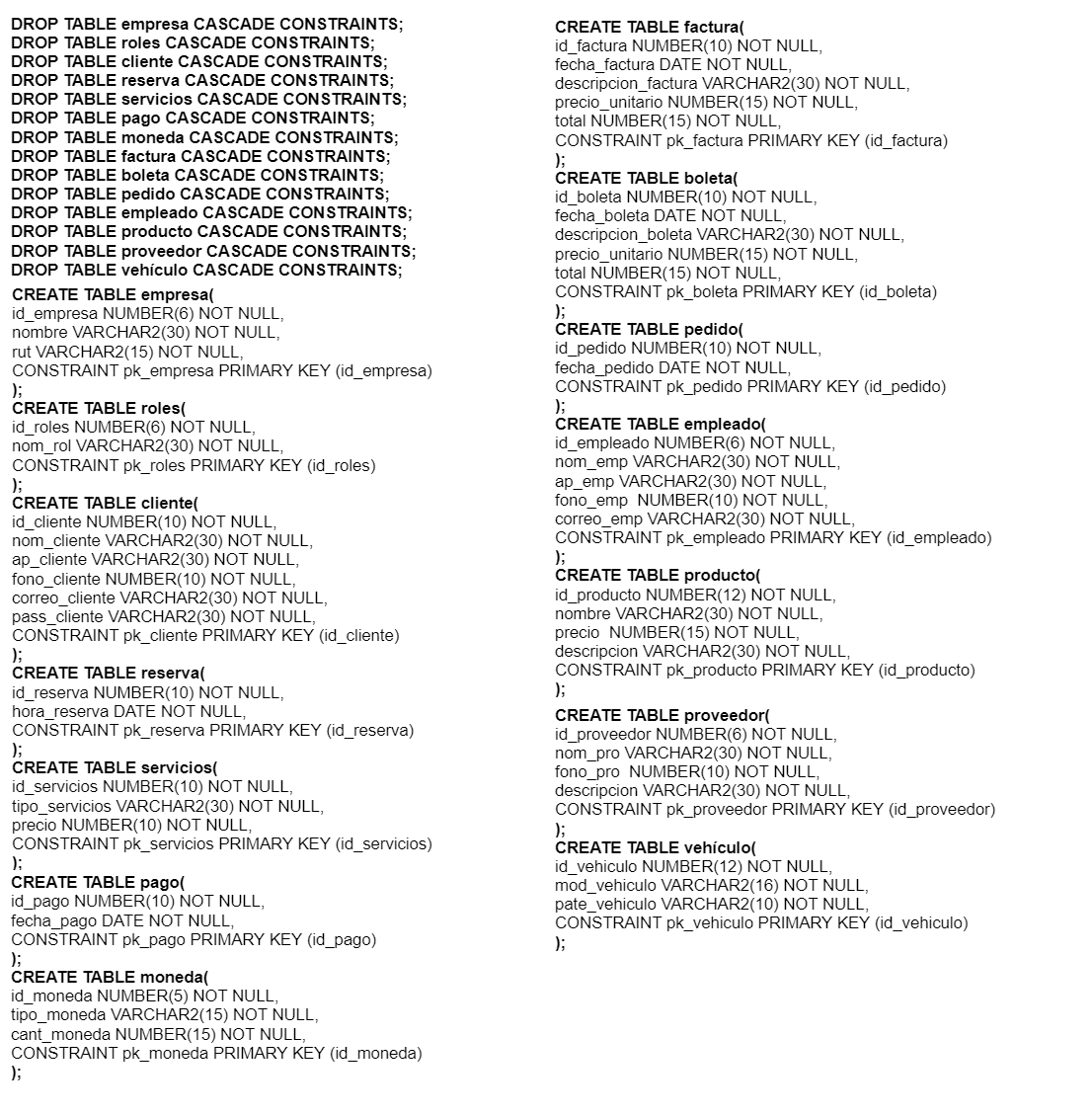
Por último, se eligió que la estrategia para implementar los providers externos en aquellos servicios asíncronos de entrada se implementarían mediante un temporizador (quartz) que levantará los procesos que verifican la llegada de información a las colas de entrada (mensajería asíncrona, archivos de texto en directorios). Esta estrategia fue seleccionada para disminuir el riesgo pues es una solución simple y efectiva.

**Anexos**

**Anexo 1 : Árbol de utilidad de la metodología ATAM**

| [**Seguridad**](#bookmark=id.gjdgxs) | *Solo 1 rol no debe estar restringido en la administración* | *(A,A) Acceso restringido a la administración del sistema,*  *según lo solicitado en seguridad y controles de acceso* |
| --- | --- | --- |
| **Usabilidad** | *El sistema permitirá usar 3 tipos de usuarios* | *(A,M) Sistema multiusuario* |
| **Adecuación**  **Funcional** | *El sistema permitirá 2 empresas relacionadas entre sí* | *(A,M )*Sistema multiempresa |
| **Adecuación**  **Funcional** | El sistema se ejecutará 3 distintas opciones de pago | *(A,M)* Sistema multi monetario |
| **Compatibilidad** | *Deben aparecer 3 opciones de descarga e impresión en distinto formato* | *(A,M)* Informes pre visualizables por pantalla, con opción impresión y exportación a otros formatos (texto, Excel, Word) |
| [**Seguridad**](#bookmark=id.gjdgxs) | *El superusuario debe tener acceso al 100% del sistema sin restricciones* | *(A,M)* El administrador del sistema es el que podrá ejecutar cambios en el mismo,debe ser tratado como súper usuario que tenga acceso a todo el sistema sin restricciones |
| **Mantenibilidad** | *El sistema se modularizada en 4 interfaces.* | *(A,M) “Módulos: Debe brindar potentes prestaciones a través de una interfaz sencilla.”* |
| **Usabilidad** | *La interfaz del sistema debe ser 100% web* | *(A,B) Un sistema de software cuya interfaz sea WEB, que permita gestionar los servicios con los distintos clientes tanto local como en el establecimiento* |
| **Usabilidad** | El manual en línea debe cubrir el 100% de las funciones del módulo | *(M,B)* Manual en línea (help) esté vinculado temáticamente al módulo que esté operando en ese momento |

**Anexo 2 : código de creación de la BD ORACLE**



**Anexo 3 : Código de creación de las clases**

| *Archivo:* ***BOLETA.py***  class BOLETA:  def \_\_init\_\_(self):  self.id\_boleta = None  self.fecha\_boleta = None  self.descripción\_boleta = None  self.precio\_unitario = None  self.total = None  self.id\_cliente = None  self.id\_producto = None  self.id\_servicio = None  self.id\_empresa = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** | *Archivo:* ***CLIENTE.py***  **class** **CLIENTE**:  **def** \_\_init\_\_(self):  self.id\_cliente = None  self.nombre = None  self.apellido = None  self.telefono = None  self.correo = None  self.pass = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** |
| --- | --- |
| *Archivo:* ***DETALLE.py***  class DETALLE:  **def** \_\_init\_\_(self):  self.Id\_detalle = None  self.precio\_unitario = None  self.cantidad = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** | *Archivo:* ***DETALLE\_PEDIDO.py***  class DETALLE\_PEDIDO:  pass |
| *Archivo:* ***EMPLEADO.py***  **class** **EMPLEADO**:  **def** \_\_init\_\_(self):  self.id\_empleado = None  self.nombre = None  self.apellido = None  self.telefono = None  self.correo = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** | *Archivo:* ***EMPRESA.py***  **class** **EMPRESA**:  **def** \_\_init\_\_(self):  self.id\_empresa = None  self.nombre = None  self.rut = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** |
| *Archivo:* ***MONEDA.py***  **class** **MONEDA**:  **def** \_\_init\_\_(self):  self.id\_moneda = None  self.tipo\_moneda = None  self.cantidad\_moneda =None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** | *Archivo:* ***FACTURA.py***  **class** **FACTURA**:  **def** \_\_init\_\_(self):  self.id\_factura = None  self.fecha\_factura = None  self.descripción\_factura = None  self.precio\_unitario = None  self.total = None  self.id\_cliente = None  self.id\_producto = None  self.id\_servicio = None  self.id\_empresa = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** |
| *Archivo:* ***PEDIDO.py***  **class** **PEDIDO**:  **def** \_\_init\_\_(self):  self.id\_pedido = None  self.fecha\_pedido = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** | *Archivo:* ***PAGO.py***  **class** **PAGO**:  **def** \_\_init\_\_(self):  self.id\_pago = None  self.fecha\_pago = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** |
| *Archivo:* ***PEDIDO\_EMP.py***  class PEDIDO\_EMP:  **def** \_\_init\_\_(self):  self.id\_pedido = None  self.fecha\_pedido = None  self.total = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** | *Archivo:* ***PRODUCTO.py***  class PRODUCTO:  **def** \_\_init\_\_(self):  self.id\_producto = None  self.nombre = None  self.precio = None  self.descripcion = None  self.Attribute2 = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** |
| *Archivo:* ***PROVEEDOR.py***  **class** **PROVEEDOR**:  **def** \_\_init\_\_(self):  self.id\_proveedor = None  self.nombre =None  self.id\_producto = None  self.descripción = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** | *Archivo:* ***RESERVAS.py***  **class** **RESERVAS**:  **def** \_\_init\_\_(self):  self.id\_reserva = None  self.hora\_reserva = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** |
| *Archivo:* ***ROLES.py***  **class** **ROLES**:  **def** \_\_init\_\_(self):  self.id\_roles = None  self.nombre = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** | *Archivo:* ***SERVICIOS.py***  **class** **SERVICIOS**:  **def** \_\_init\_\_(self):  self.id\_servicio = None  self.nombre = None  self.precio = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** |
| *Archivo:* ***VEHICULO.py***  **class** **VEHICULO**:  **def** \_\_init\_\_(self):  self.id\_vehiculo = None  self.modelo = None  self.patente = None  **def** Ingresar(self, ):  **pass**  **def** Modificar(self, ):  **pass**  **def** Consultar(self, ):  **pass**  **def** Eliminar(self, ):  **pass** |  |

* *Todo este código fue generado con la extensión de* ***StarUML*** *especializada en* ***Python****.*