PARRILLA SHORI

**Documento de Especificación de Arquitectura**

Realizado por:

Adrian Mojica, Omar Quinchoa , Mauro Martínez y Juan Medina

***HISTORIAL DE REVISIONES***

| **Fecha** | **Versión** | **Autor** | **Descripción** | **Revisado Por** |
| --- | --- | --- | --- | --- |
| 20/06/2025 | 1.1 | Adrian Mojica | Segundo Avance |  |

**Contenido**

[1.](#_heading=h.3znysh7) Documento de Arquitectura de Software 4

[1.1.](#_heading=h.2et92p0) Introducción 4

[1.2.](#_heading=h.tyjcwt) Propósito 4

[1.3.](#_heading=h.3dy6vkm) Alcance 4

[1.4.](#_heading=h.1t3h5sf) Referencias 4

[1.5.](#_heading=h.4d34og8) Definiciones acrónimos y abreviaciones 4

[2.](#_heading=h.2s8eyo1) Generalidades del Proyecto 5

[2.1.](#_heading=h.17dp8vu) Problema a Resolver 5

[2.2.](#_heading=h.3rdcrjn) Descripción General del Sistema a Desarrollar 5

[2.3.](#_heading=h.26in1rg) Identificación de los Stakeholders y sus responsabilidades 5

[3.](#_heading=h.lnxbz9) Vistas de la arquitectura 5

[3.1.](#_heading=h.35nkun2) Vista de Casos de Uso 5

[3.2.](#_heading=h.1ksv4uv) VISTA DE PROCESOS 5

[3.3.](#_heading=h.44sinio) VISTA LÓGICA 5

[3.4.](#_heading=h.z337ya) VISTA DE IMPLEMENTACIÓN 6

[3.5.](#_heading=h.3j2qqm3) VISTA DE DESPLIEGUE 6

[4.](#_heading=h.1y810tw) Arquitectura en capas 6

[5.](#_heading=h.4i7ojhp) VISTA DE DATOS 6

[5.1.](#_heading=h.2xcytpi) Modelo Relacional 6

[6.](#_heading=h.1ci93xb) Definición de Interfaces de Usuario 6

[7.](#_heading=h.3whwml4) Características Generales de Calidad 6

[7.1.](#_heading=h.2bn6wsx) Tamaño y performance 6

[7.2.](#_heading=h.1pxezwc) Calidad 6

[7.3.](#_heading=h.49x2ik5) Usabilidad 7

[7.4.](#_heading=h.2p2csry) Eficiencia 7

[7.5.](#_heading=h.147n2zr) Seguridad 7

[7.6.](#_heading=h.3o7alnk) Confiabilidad 7

[7.7.](#_heading=h.23ckvvd) Mantenimiento 7

[7.8.](#_heading=h.ihv636) Estándares 7

1. **Documento de Arquitectura de Software**
   1. **Introducción**

El presente documento de Especificación de Arquitectura de Software tiene como propósito describir de manera estructurada la arquitectura del sistema “Shori Express”, una solución tecnológica diseñada para optimizar los procesos operativos y comerciales del restaurante Parrilla Shori.

Este sistema surge como respuesta a diversas problemáticas identificadas en la gestión interna del establecimiento, particularmente en lo relacionado con el manejo de inventarios y la ausencia de un canal digital para la realización de pedidos a domicilio y su respectiva facturación. Así mismo, busca potenciar la relación con los clientes mediante un sistema de fidelización basado en recompensas.

En este documento se detallan las vistas arquitectónicas del sistema, su estructura lógica y física, el modelo de datos, los componentes de software, las interfaces de usuario, y los atributos de calidad que aseguran la eficiencia, usabilidad, seguridad y mantenibilidad del producto. A través de esta especificación, se busca garantizar una comprensión clara y compartida del sistema a desarrollar por parte de todos los involucrados en el proceso de análisis, diseño, desarrollo e implementación.

* 1. **Propósito**

El propósito de este documento es definir y comunicar la arquitectura del sistema de software “Shori Express”, desarrollado para la empresa Parrilla Shori. Esta especificación está dirigida a todos los interesados en el ciclo de vida del proyecto (analistas, desarrolladores, testers, clientes, entre otros) y tiene como finalidad proporcionar una guía clara y detallada de la estructura técnica del sistema.

Este documento sirve como una base común para comprender el funcionamiento interno del sistema, facilitar el desarrollo modular, permitir el mantenimiento a futuro y garantizar que los requisitos funcionales y no funcionales del proyecto se vean reflejados de forma coherente en la arquitectura del software.

* 1. **Alcance**

La arquitectura definida en este documento abarca el desarrollo de una solución digital integral que incluye dos módulos principales: uno administrativo para la gestión interna del restaurante y otro comercial orientado al cliente final.

El sistema contempla funcionalidades como el control de inventarios, alertas por bajo stock, visualización de menú, gestión de pedidos a domicilio, opciones de pago, seguimiento del estado del pedido y un sistema de fidelización mediante bonos por compras recurrentes. Adicionalmente, se incluye un panel centralizado para la visualización de estadísticas, reportes y gestión de clientes.

En su primera versión, el alcance del sistema excluye la integración con plataformas externas de domicilios (como Rappi o Uber Eats), el desarrollo de aplicaciones móviles nativas y la inclusión de módulos de facturación electrónica, salvo que se requieran como componentes adicionales en futuras fases del proyecto.

* 1. **Referencias**

1. https://docs.google.com/spreadsheets/d/1f-HtEEju-RSVX2c\_cLX0j1nRmgeIaV1m/edit?usp=sharing&ouid=115136602959028200183&rtpof=true&sd=true
2. Documento de Visión del Proyecto.

https://docs.google.com/document/d/1yo5GJ6ev4NY1G-JYqQKcaTapDrJdswb1/edit?usp=sharing&ouid=115136602959028200183&rtpof=true&sd=true

1. Planilla de análisis del software.
   1. **Definiciones acrónimos y abreviaciones**

* **Arquitectura de Software**: Es el diseño estructural de un sistema de software. Define los componentes principales, sus relaciones, interacciones y principios que guían su diseño y evolución. Establece la base técnica sobre la cual se construye el sistema.
* **Descripción de Arquitectura**: Es una representación documentada que explica cómo está organizada la arquitectura de un sistema. Incluye vistas, decisiones técnicas, componentes, relaciones, tecnologías utilizadas y cómo cumplen con los requisitos funcionales y no funcionales.
* **Vista**: Una vista es una representación específica de una parte del sistema enfocada en ciertas preocupaciones, como estructura lógica, comportamiento, despliegue, etc. Cada vista muestra una perspectiva particular del sistema para ciertos stakeholders.
* **Tipo de Vista**: Convención que indica cómo se debe construir y documentar una vista específica de la arquitectura.
* **Stakeholder**:Persona, grupo u organización que tiene interés o se ve afectado por el sistema. Ejemplos: usuarios finales, clientes, desarrolladores, testers, administradores del sistema.
* **Inventario**: Conjunto de productos o recursos almacenados que están disponibles para la venta, uso o distribución. En software, es común representar el inventario en módulos de gestión de existencias.
* **Stock**: Cantidad específica de productos disponibles en el inventario. El sistema suele controlarlo para evitar desabastecimientos o sobrestock.
* **Pedido a Domicilio**: Funcionalidad que permite a los usuarios solicitar productos o servicios para ser entregados en su ubicación. Generalmente implica una interfaz para ordenar, pago en línea y gestión de entregas.
* **Sistema de Fidelización**: Módulo que premia a los clientes frecuentes con descuentos, puntos, recompensas u otros beneficios para fomentar la lealtad y repetición de compras.
* **Panel Administrativo**: Interfaz destinada a los administradores del sistema para gestionar datos, usuarios, productos, pedidos, reportes, etc. Suele ser de acceso restringido.
* **Frontend**: Parte visible del sistema con la que interactúa el usuario final. Incluye interfaces gráficas, formularios, navegación, etc. Se desarrolla con tecnologías como HTML, CSS, JavaScript y frameworks como React o Vue.
* **Backend**:Parte del sistema que se encarga de la lógica de negocio, procesamiento de datos, autenticación y conexión con bases de datos. Suele estar desarrollado con lenguajes como PHP, Python, Node.js, Java, etc.
* **Base de Datos (MySQL)**: Sistema de gestión de base de datos relacional que permite almacenar y consultar datos estructurados. MySQL es uno de los sistemas más populares, especialmente para aplicaciones web.
* **UI (User Interface)**: Interfaz de Usuario: conjunto de elementos gráficos y visuales que permiten la interacción entre el usuario y el software (botones, menús, formularios, íconos, etc.).
* **Responsive**: Diseño adaptable que permite que una interfaz se vea y funcione correctamente en diferentes dispositivos y tamaños de pantalla (móvil, tablet, escritorio).
* **CRUD**: Acrónimo de **Create, Read, Update, Delete**, que representa las operaciones básicas de gestión de datos que el sistema permite realizar sobre los distintos módulos.

1. **Generalidades del Proyecto**
   1. **Problema a Resolver**

La empresa **“Parrilla Shori”**, un establecimiento de comida especializado en carnes a la parrilla, ha venido enfrentando diversos desafíos operativos que afectan su eficiencia, control administrativo y competitividad. Uno de los principales problemas radica en la **falta de organización en la gestión de inventarios**, lo cual ha generado pérdidas de productos, errores en el control de insumos y dificultades para planificar compras y abastecimiento. Esta situación impacta directamente en la calidad del servicio y la rentabilidad del negocio.

Adicionalmente, la **ausencia de un canal de domicilios estructurado y automatizado** limita el alcance comercial del restaurante, impidiendo que los clientes puedan hacer pedidos de manera cómoda y rápida desde sus hogares o lugares de trabajo. En un entorno altamente competitivo y digitalizado, esta carencia representa una desventaja significativa frente a otras marcas del sector gastronómico.

Otro aspecto crítico es la **falta de un sistema de facturación digital**. Actualmente, la emisión de comprobantes de pago se realiza de forma manual o poco sistematizada, lo que puede derivar en errores contables, duplicación de datos, dificultades en el control financiero y falta de trazabilidad entre los pedidos y sus respectivos pagos. La inexistencia de facturas automáticas también complica la transparencia y la relación formal con el cliente.

Con el objetivo de transformar estas debilidades en oportunidades, surge la necesidad de desarrollar el proyecto **Shori Express**, una plataforma digital integral que permitirá optimizar los procesos internos, especialmente el manejo de inventario y la facturación, al tiempo que habilita un sistema de pedidos a domicilio eficiente e intuitivo para los clientes.

Como valor agregado, se plantea también la implementación de un **programa de fidelización**, mediante el cual los clientes frecuentes recibirán un bono o recompensa después de acumular un número determinado de compras, incentivando así la lealtad, el retorno continuo y la recomendación del servicio.

En este contexto, se vuelve imprescindible diseñar e implementar una **solución tecnológica completa**, que mejore la organización operativa, formalice el proceso de ventas mediante la facturación automática, amplíe el alcance comercial y fortalezca la relación con los clientes actuales y potenciales.

* 1. **Descripción General del Sistema a Desarrollar (General y por módulo)**

Desarrollar e implementar una solución digital para la Parrilla Shori llamada Shori Express, que permita mejorar el control de inventarios, generar facturas de forma automática, ofrecer pedidos a domicilio en línea y premiar a los clientes frecuentes con bonos por sus compras.

1. Módulo de Gestión Interna (Administrativo)

* Registro, actualización y control de **inventario de productos e insumos**.
* Notificaciones de **bajo stock o vencimiento** de productos.
* Registro de **entradas y salidas de inventario** según compras o ventas.
* Generación de reportes básicos sobre **movimientos de inventario**.

2. Módulo Comercial (Cliente / Punto de Venta Digital)

* Interfaz web o móvil para que los clientes puedan:
  + Visualizar el **menú actualizado** con precios y descripciones.
  + Realizar **pedidos para domicilio** seleccionando productos y cantidades.
  + Elegir opciones de **pago (efectivo, transferencia o plataformas integradas)**.
* Confirmación y seguimiento del estado del pedido (**en preparación**, **en camino**, **entregado**).

3. Sistema de Fidelización

* Registro automático de las compras por cliente.
* Acumulación de puntos o historial de compras.
* Generación de **bonos o recompensas** al alcanzar un número definido de pedidos.
* Notificación automática al cliente sobre el estado de su bono.

4. Panel Administrativo Centralizado

* Visualización y gestión de:
  + Pedidos en curso y finalizados.
  + Inventario y alertas.
  + Datos de clientes frecuentes y estadísticas de fidelización.
* Reportes generales de ventas, productos más pedidos y análisis de comportamiento del cliente.

5. Módulo de Facturación

* Generación automática de facturas al momento de finalizar un pedido.
* Registro detallado de productos, cantidades, precios, subtotal, IVA y total.
* Asociación directa de la factura con el pedido correspondiente.
* Almacenamiento y consulta de facturas para el cliente y el administrador.
* Opción de descarga en PDF e impresión.
* Configuración de parámetros de facturación desde el panel administrativo (IVA, encabezado, numeración, etc.).

Límites del alcance (exclusiones iniciales)

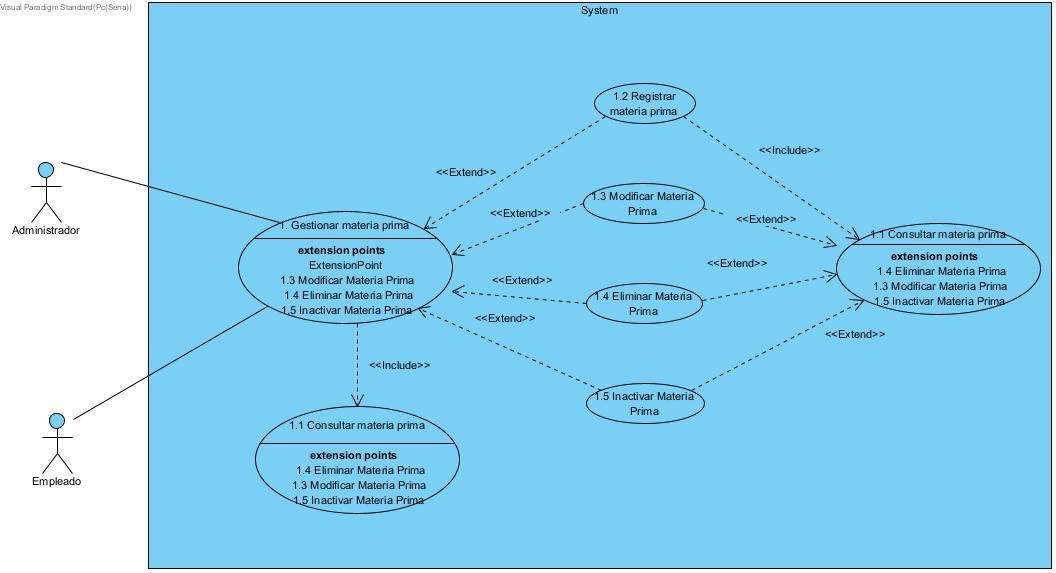
* El sistema no contempla integración con plataformas externas (como Rappi o Uber Eats) en su fase inicial.
* No se incluye una aplicación móvil nativa (App Store / Play Store), aunque puede diseñarse una versión web adaptable (responsive).
* El control de caja y facturación electrónica no está incluido a menos que se solicite como un módulo adicional.
  1. **Identificación de los Stakeholders y sus responsabilidades**

| **STAKEHOLDER** | **DESCRIPCIÓN** | **ESCENARIO** | **CASO DE USO** |
| --- | --- | --- | --- |
| Admin | Usuario con permisos totales en el sistema. | Cuando necesite ver reportes o gestionas usuarios | Ver reportes / Crear usuarios / Dar permisos |
| Empleado | Mesero o persona que atiende | Cuando toma pedidos o consulta lo que hay en el inventario | Registrar pedido / Consultar stock |
| Cliente | La persona que llega a la | Cuando hace un pedido o está esperando ser atendido | Ver el estado de su pedido |

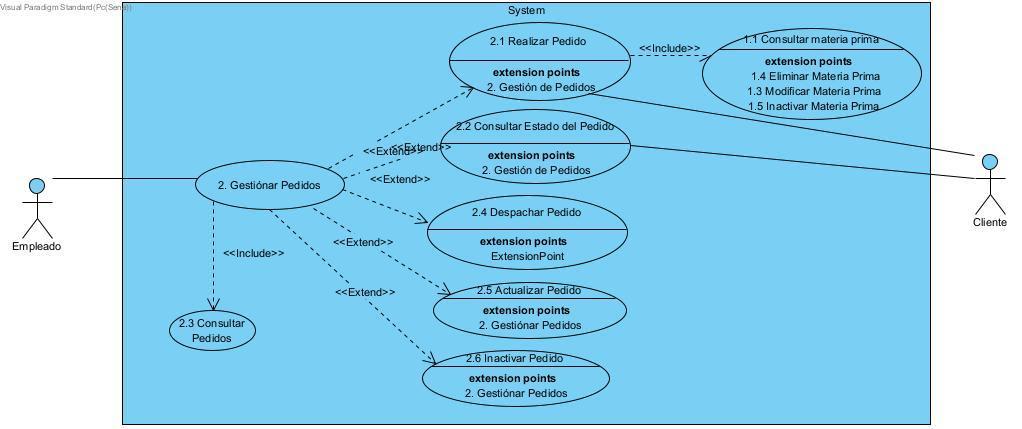
1. **Vistas de la arquitectura**
   1. **Vista de Casos de Uso (Se debe colocar cada uno de los casos de uso que han sido elaborados y requeridos para el software)**

Para cada caso de uso, se debe argumentar como valor agregado a la comprensión del mismo (imagen y observaciones importante a tener en cuenta).

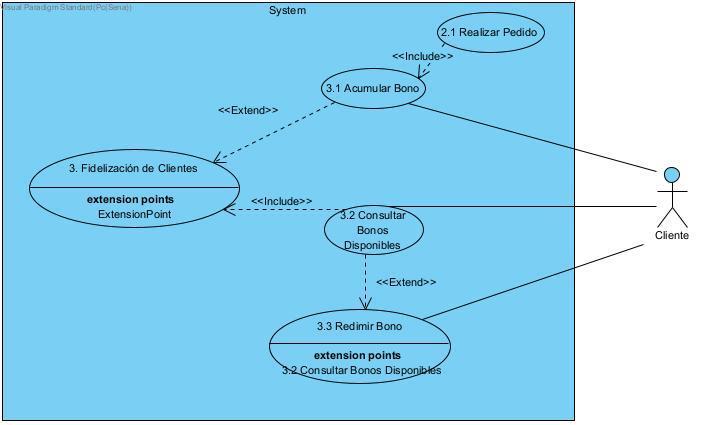
Materia prima



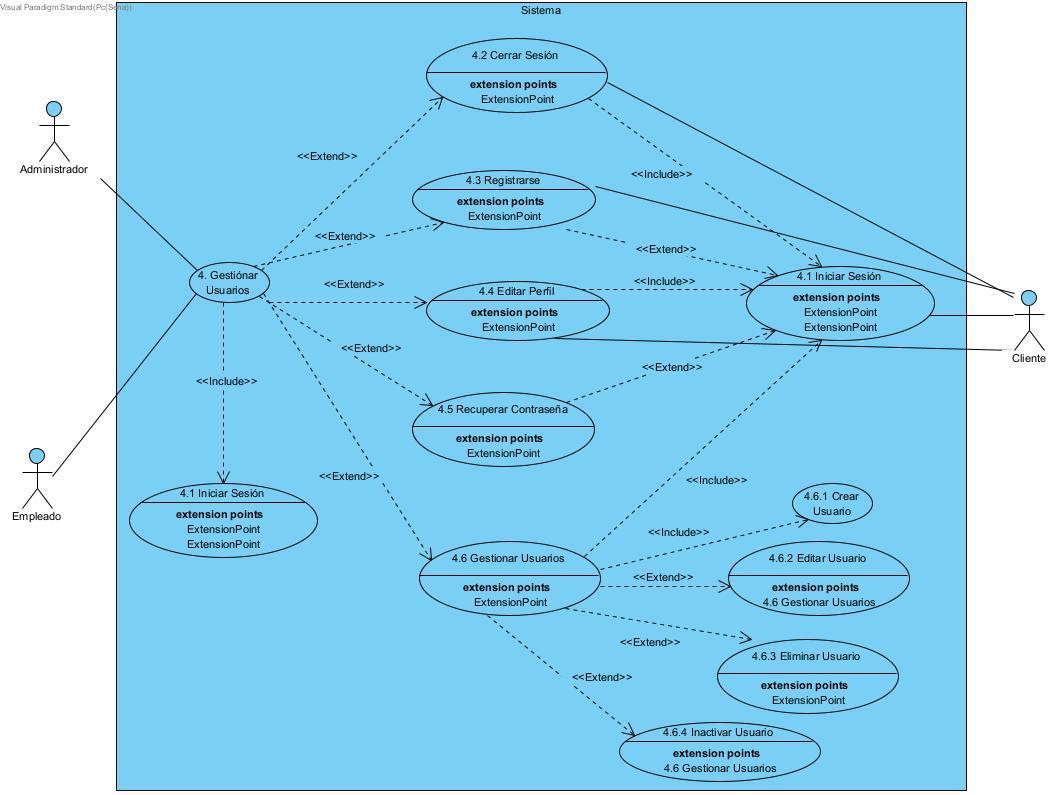
Pedido



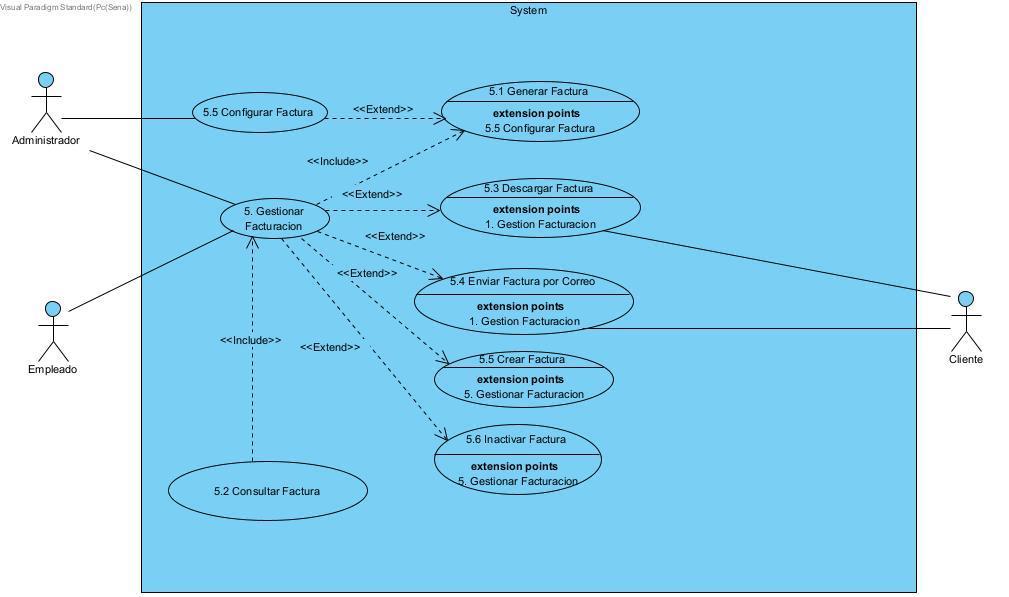
Acumular bono



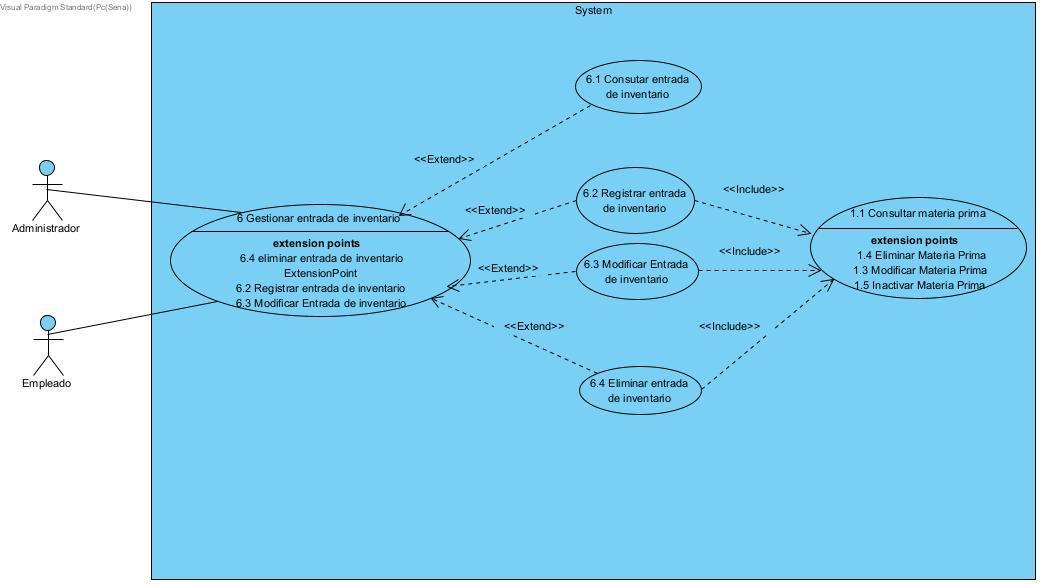
Iniciar Sesión



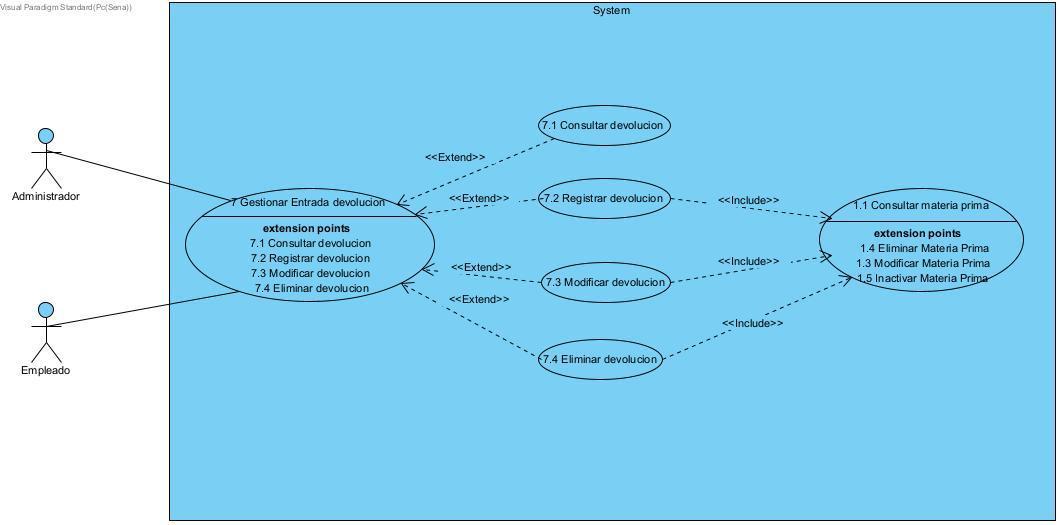
Generar factura



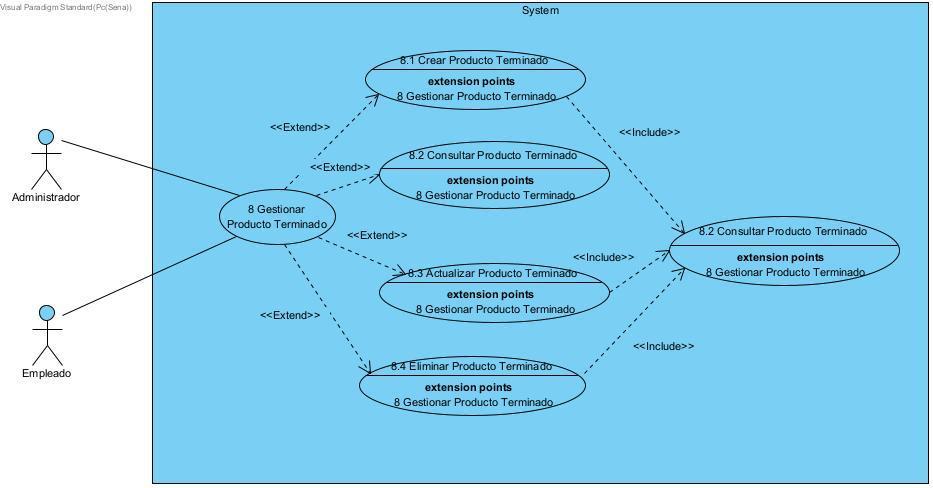
Entrada inventario



Devolución



producto terminado



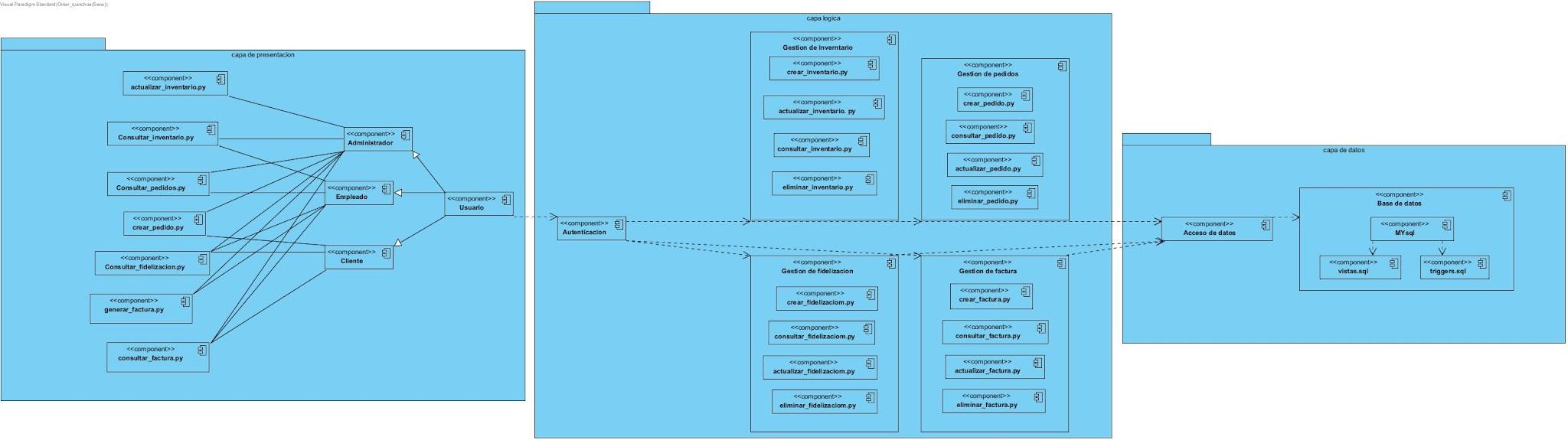
* 1. **VISTA DE PROCESOS**

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.

* + 1. **Diagrama de Actividades**
  1. **VISTA LÓGICA**
     1. **Diagramas - Clases (Actualizado)**
     2. **Diagramas - Secuencia**

Los diagramas de los puntos 3.4 y 3.5.1 corresponden a los diagramas que se han elaborado en la fase III

* 1. **VISTA DE IMPLEMENTACIÓN**
     1. **Diagrama de Componentes**

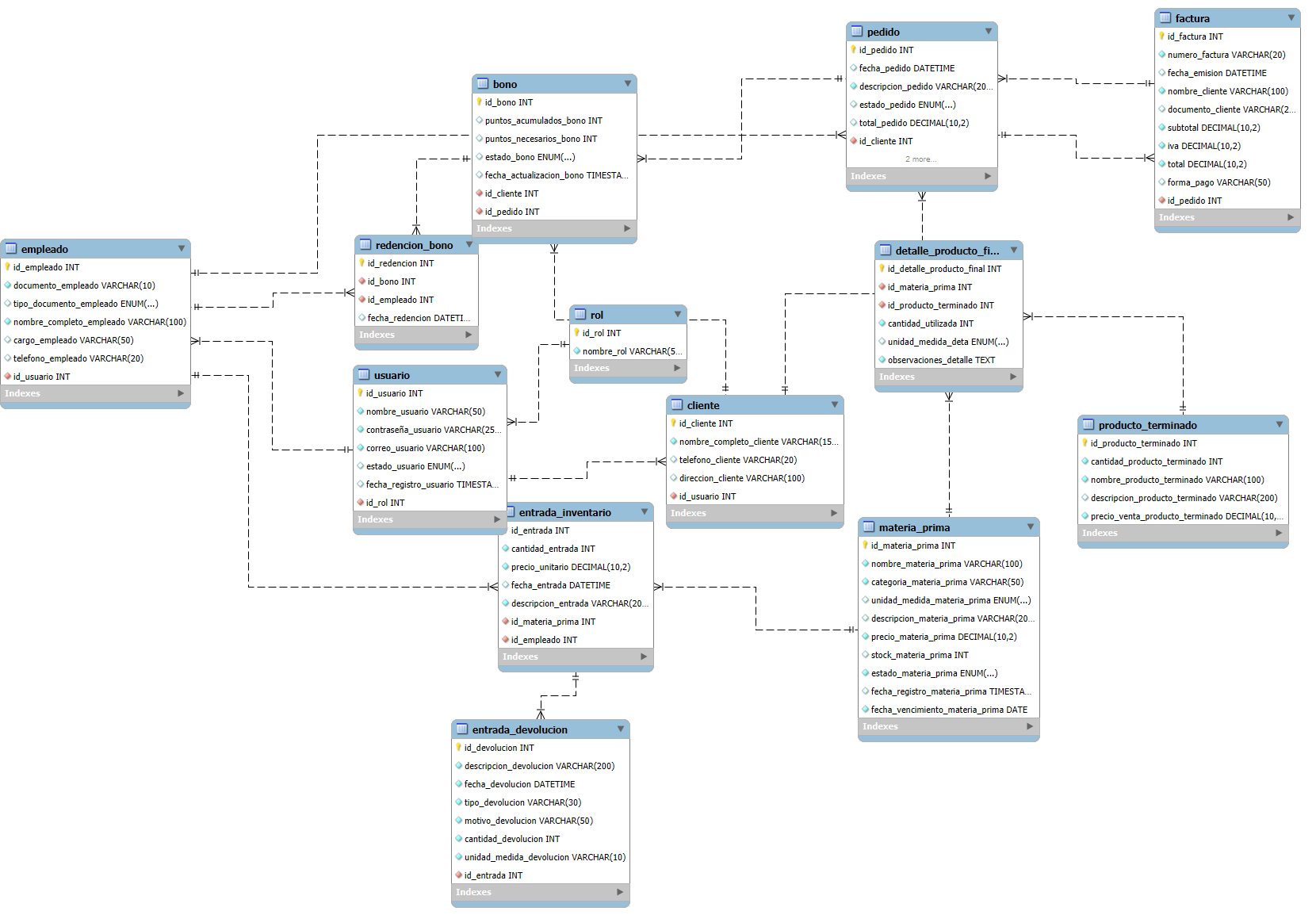


* + 1. **Diagrama de Paquetes**
  1. **VISTA DE DESPLIEGUE**
     1. **Diagrama de despliegue**

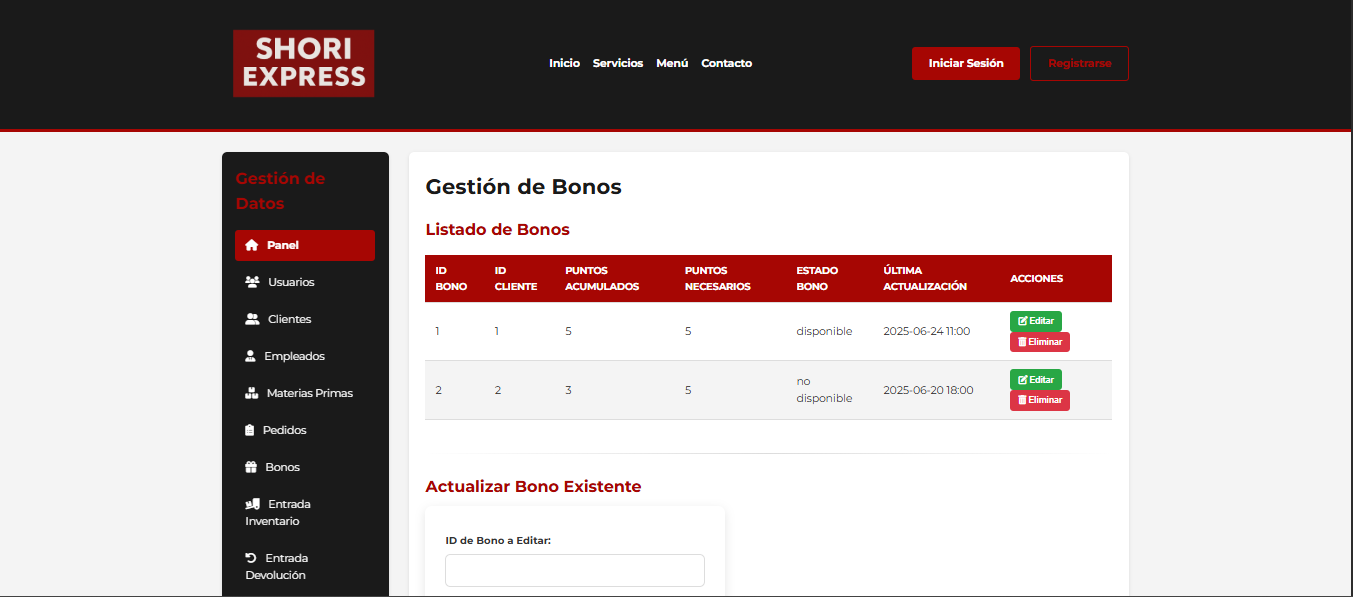
1. **Arquitectura en capas**

* Cuadro de Tecnologías por capas- versiones

1. **VISTA DE DATOS**
   1. Modelo Relacional normalizado tercera forma normal de la base de datos (SGBD).



1. **Definición de Interfaces de Usuario**



* Página principal, formularios, páginas error.
* Landing Page (Página Principal).
* Pantallazos interfaces modulares.

1. **Características Generales de Calidad**
   1. **Tamaño y performance**

* 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.

* 1. **Calidad**
  2. **Usabilidad**
  3. **Eficiencia**
  4. **Seguridad**
  5. **Confiabilidad**
  6. **Mantenimiento**
  7. **Estándares**

Normas de calidad, mencionar dos normas de calidad de Desarrollo de Software y argumentar cómo esas normas aplican para el software.