

ERP logística de eventos

Estudiosos:

Santiago Castro González

Anderson Yecid Hurtado Puentes

Victor Alfonso Alarcón Ospina

ARQUITECTURA DEL SOFTWARE

Ingeniería en software

Universidad Manuela Beltrán

13/2/2025

**ERP Logístico - Documentación en Formato arc42**

1. **Introducción y Objetivos (Introduction and Goals)**

**1.1 Descripción General**

El ERP Logístico es un sistema integral diseñado para mejorar la gestión de empresas del sector logístico. Su propósito es automatizar procesos clave, optimizar recursos y facilitar la toma de decisiones basadas en datos confiables.

Este ERP no solo es una herramienta funcional, sino una plataforma escalable y segura que se adapta al crecimiento del negocio. Su diseño garantiza una experiencia de usuario intuitiva, brindando acceso desde computadoras, tablets y dispositivos móviles.

**1.2 Objetivos de Negocio**

El ERP Logístico busca resolver los desafíos diarios de una empresa del sector mediante la digitalización y automatización de sus procesos. Sus principales beneficios incluyen:

* Mayor eficiencia operativa: Menos errores manuales y procesos más ágiles.
* Mejor control financiero: Gestión precisa de ingresos, egresos y cuentas por pagar/cobrar.
* Excelencia en el servicio al cliente: Comunicación fluida y herramientas digitales modernas.
* Toma de decisiones basada en datos: Reportes avanzados y dashboards personalizables.
* Seguridad y confianza: Protección de datos con cifrado y auditorías.

**1.3 Características Esenciales**

El ERP Logístico destaca por su diseño modular y flexible, permitiendo a las empresas personalizar su uso según sus necesidades. Sus principales características incluyen:

* Modularidad y escalabilidad: Crece con el negocio sin afectar el rendimiento.
* Interfaz amigable: Adaptable a múltiples dispositivos con una navegación intuitiva.
* Seguridad avanzada: Autenticación, cifrado y auditoría de registros.
* Integración con sistemas externos: API REST para conectar con e-commerce y software de terceros.
* Reportes personalizados: Exportación en PDF, Excel y gráficos interactivos.

**1.4 Requerimientos Funcionales**

El ERP abarca todas las áreas clave de una empresa logística:

**Recursos Humanos**

* Registro y gestión de empleados.
* Control de asistencia y jornadas laborales.
* Administración de roles y permisos.
* Automatización de nómina y pagos.

**Inventario**

* Gestión de productos, equipos y materiales.
* Registro de movimientos (entradas/salidas).
* Alertas automáticas para reabastecimiento.
* Asignación de recursos a empleados y servicios.

**Proveedores**

* Registro y administración de proveedores.
* Creación de órdenes de compra y pagos.
* Control de recepción de insumos y reportes de incidencias.
* Evaluación del desempeño de proveedores.

**Clientes**

* Gestión de datos, contratos y preferencias.
* Administración de pedidos y facturación.
* Registro de pagos y acceso al historial de transacciones.
* Integración con WhatsApp y correos automáticos para notificaciones.

**Contabilidad**

* Control de ingresos y egresos.
* Administración de cuentas por pagar y cobrar.
* Generación de reportes financieros y presupuestarios.
* Cumplimiento de normativas fiscales.

**Servicios**

* Programación y seguimiento de servicios logísticos.
* Asignación de empleados y recursos.
* Control de costos y facturación automática.
  1. **Objetivos de Calidad**

Para garantizar el éxito del ERP Logístico, se han definido los siguientes estándares de calidad:

**Rendimiento y escalabilidad**

* Tiempo de respuesta menor a 2 segundos en operaciones críticas.
* Soporte para más de 100 usuarios simultáneos sin pérdida de rendimiento.
* Arquitectura modular para futuras expansiones.

**Seguridad**

* Autenticación y autorización basada en roles.
* Cifrado de datos en tránsito y almacenamiento.
* Auditoría de registros para rastrear cambios y detectar anomalías.

**Usabilidad y experiencia de usuario**

* Interfaz intuitiva y adaptable a cualquier dispositivo.
* Dashboard personalizable con métricas clave.
* Soporte multilenguaje para clientes internacionales.

**Integración y compatibilidad**

* API REST documentada para facilitar conexiones con otros sistemas.
* Compatible con navegadores modernos y móviles.
* Reportes en formatos estándar como PDF y Excel.

**Mantenimiento y soporte**

* Documentación clara para usuarios y desarrolladores.
* Alertas proactivas para detectar problemas antes de afectar el servicio.
* Arquitectura basada en microservicios para fácil mantenimiento.

**1.6 Partes Interesadas y Expectativas**

Cada actor dentro del ERP Logístico tiene sus propias necesidades y expectativas:

|  |  |
| --- | --- |
| **Parte Interesada** | **Expectativas** |
| Gerencia | Control total de operaciones mediante reportes claros y dashboards estratégicos. |
| Empleados | Acceso rápido a información relevante, gestión de asistencia y pagos puntuales. |
| Clientes | Servicio eficiente, facturación digital y comunicación fluida. |
| Proveedores | Procesamiento ágil de órdenes de compra y pagos automatizados. |
| Equipo de TI | Arquitectura modular, APIs bien definidas y documentación clara. |

1. **Visión General de Requisitos (Requirements Overview)**

**2.1 Descripción Breve de los Requisitos Funcionales**

El **ERP Logístico** es un sistema diseñado para la gestión eficiente de empresas del sector logístico. Su objetivo principal es **automatizar, optimizar y digitalizar procesos clave** como la gestión de recursos humanos, inventarios, proveedores, clientes, contabilidad y servicios.

Este ERP está basado en una arquitectura modular y escalable, lo que permite su **adaptabilidad a distintos tamaños de empresas y necesidades específicas** dentro del sector logístico.

**Versiones y Ubicación de la Documentación:**

* Documento de requisitos (Versión 1.0).
* Ubicación: Repositorio interno del proyecto / Documentación en la nube.

**2.2 Motivación**

Desde el punto de vista de los usuarios finales, este sistema se desarrolla para **mejorar la eficiencia en la operación logística y optimizar la calidad de los servicios**.

Algunos de los principales problemas que busca resolver incluyen:

* **Procesos manuales ineficientes** que generan errores y retrasos.
* **Falta de control financiero y contable** en la administración de recursos.
* **Desorden en la gestión de inventarios y proveedores**, dificultando la planificación de insumos.
* **Deficiencias en la comunicación con clientes y proveedores**, impactando la satisfacción del servicio.
* **Dificultad en la toma de decisiones basada en datos**, debido a la falta de informes centralizados y analíticos.

Con este ERP, se pretende lograr una **mayor automatización, integración y digitalización** de los procesos, permitiendo a la empresa operar con **mayor rapidez, eficiencia y seguridad**.

**2.3 Forma y Presentación de los Requisitos**

A continuación, se presenta una descripción en formato tabular de los principales casos de uso y requisitos funcionales del ERP Logístico:

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Módulo** | **Funcionalidad** | **Descripción** |
| RF-01 | Recursos Humanos | Gestión de empleados | Registro y administración de empleados, roles y permisos. |
| RF-02 | Recursos Humanos | Control de asistencia | Registro de entrada/salida y generación de reportes de asistencia. |
| RF-03 | Inventario | Control de existencias | Registro de stock, alertas de reabastecimiento y movimientos de inventario. |
| RF-04 | Inventario | Asignación de recursos | Permite vincular recursos específicos a empleados o servicios. |
| RF-05 | Proveedores | Gestión de proveedores | Registro de proveedores, contratos y evaluaciones de rendimiento. |
| RF-06 | Clientes | Gestión de clientes | Registro de clientes, preferencias y contratos de servicio. |
| RF-07 | Contabilidad | Control financiero | Administración de ingresos, egresos y generación de reportes financieros. |
| RF-08 | Servicios | Planificación de operaciones | Programación y asignación de servicios logísticos. |
| RF-09 | Seguridad | Control de acceso | Implementación de roles y permisos basados en jerarquía. |
| RF-10 | Integración | API REST | Permite la conexión con sistemas externos, como tiendas online o CRMs. |

* 1. **Referencias y Documentación Relacionada**

Este documento es una extracción de la documentación general del **ERP Logístico** bajo el formato **arc42**. Para más detalles sobre los requisitos y especificaciones, consultar:

* Documento completo de Requisitos Funcionales (Versión 1.0).
* Especificaciones técnicas del sistema.
* Diagramas de arquitectura y casos de uso.
  1. **Objetivos de Calidad (Quality Goals)**

**Contenido**

Este apartado define los tres a cinco principales objetivos de calidad para la arquitectura, cuya consecución es de máxima importancia para los principales interesados. Estos objetivos están alineados con el estándar ISO 25010 y tienen un impacto directo en las decisiones arquitectónicas clave.

**Motivación**

Los objetivos de calidad deben estar claramente definidos para garantizar que la arquitectura del sistema cumpla con las expectativas de los interesados. Estos objetivos influyen en decisiones fundamentales del diseño y ayudan a priorizar aspectos clave en el desarrollo. Es esencial que los arquitectos comprendan cómo será evaluada la calidad de su trabajo.

**Objetivos de Calidad**

A continuación, se presenta una tabla con los principales objetivos de calidad, su descripción y escenarios concretos donde se aplican, ordenados por prioridad.

|  |  |  |
| --- | --- | --- |
| **Prioridad** | **Objetivo de Calidad** | **Descripción** |
| 1 | Seguridad | Garantizar la protección del sistema contra accesos no autorizados, modificaciones no deseadas y ataques maliciosos. Implementación de cifrado, autenticación y control de accesos. |
| 2 | Fiabilidad | Asegurar que el sistema mantenga un nivel estable de rendimiento y disponibilidad bajo condiciones operativas definidas. Implementación de estrategias de redundancia y recuperación ante fallos. |
| 3 | Mantenibilidad | Facilitar la actualización, corrección y adaptación del sistema a cambios en los requisitos o el entorno. Uso de una arquitectura modular y documentación clara. |
| 4 | Compatibilidad | Permitir la interoperabilidad con otros sistemas y plataformas. Uso de estándares abiertos y APIs bien definidas. |
| 5 | Usabilidad | Asegurar que el sistema sea intuitivo, fácil de aprender y atractivo para los usuarios. Diseño centrado en el usuario y pruebas de usabilidad continuas. |

**3. Visión General de requerimientos**

**3.1. Contenido**

El ERP Logístico es un sistema diseñado para la gestión eficiente de empresas del sector logístico. Su objetivo principal es automatizar, optimizar y digitalizar procesos clave como la gestión de recursos humanos, inventarios, proveedores, clientes, contabilidad y servicios.

Este ERP está basado en una arquitectura modular y escalable, lo que permite su adaptabilidad a distintos tamaños de empresas y necesidades específicas dentro del sector logístico.

**Versiones y Ubicación de la Documentación:**

* Documento de requisitos (Versión 1.0).
* Ubicación: Repositorio interno del proyecto / Documentación en la nube.

**3.2. Motivación**

Desde el punto de vista de los usuarios finales, este sistema se desarrolla para mejorar la eficiencia en la operación logística y optimizar la calidad de los servicios.

Algunos de los principales problemas que busca resolver incluyen:

* Procesos manuales ineficientes que generan errores y retrasos.
* Falta de control financiero y contable en la administración de recursos.
* Desorden en la gestión de inventarios y proveedores, dificultando la planificación de insumos.
* Deficiencias en la comunicación con clientes y proveedores, impactando la satisfacción del servicio.
* Dificultad en la toma de decisiones basada en datos, debido a la falta de informes centralizados y analíticos.

Con este ERP, se pretende lograr una mayor automatización, integración y digitalización de los procesos, permitiendo a la empresa operar con mayor rapidez, eficiencia y seguridad.

**3.3. Forma y Presentación de los Requerimientos**

A continuación, se presenta una descripción en formato tabular de los principales casos de uso y requisitos funcionales del ERP Logístico:

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Módulo** | **Funcionalidad** | **Descripción** |
| RF-01 | RRHH | Gestión de empleados | Registro y administración de empleados, roles y permisos. |
| RF-02 | RRHH | Control de asistencia | Registro de entrada/salida y generación de reportes de asistencia. |
| RF-03 | Inventario | Control de existencias | Registro de stock, alertas de reabastecimiento y movimientos de inventario. |
| RF-04 | Inventario | Asignación de recursos | Permite vincular recursos específicos a empleados o servicios. |
| RF-05 | Proveedores | Gestión de proveedores | Registro de proveedores, contratos y evaluaciones de rendimiento. |
| RF-06 | Clientes | Gestión de clientes | Registro de clientes, preferencias y contratos de servicio. |
| RF-07 | Contabilidad | Control financiero | Administración de ingresos, egresos y generación de reportes financieros. |
| RF-08 | Servicios | Planificación de operaciones | Programación y asignación de servicios logísticos. |
| RF-09 | Seguridad | Control de acceso | Implementación de roles y permisos basados en jerarquía. |
| RF-10 | Integración | API REST | Permite la conexión con sistemas externos, como tiendas online o CRMs. |

* 1. **Interesados (Stakeholders)**

**Contenido**

Este apartado proporciona una visión general explícita de los interesados en el sistema, incluyendo todas las personas, roles u organizaciones que:

* Deben conocer la arquitectura del sistema.
* Necesitan ser convencidos sobre la viabilidad de la arquitectura.
* Trabajarán directamente con la arquitectura o el código.
* Requieren la documentación de la arquitectura para su labor.
* Deben tomar decisiones sobre el sistema o su desarrollo.

**Motivación**

Es fundamental identificar todas las partes involucradas en el desarrollo del sistema o aquellas que serán impactadas por él. De lo contrario, podrían surgir problemas inesperados durante el proceso de desarrollo. Estos interesados determinan el alcance y el nivel de detalle requerido en el trabajo y los resultados de la arquitectura.

**Forma**

A continuación, se presenta una tabla con los nombres de los roles, contactos y sus expectativas con respecto a la arquitectura y su documentación.

|  |  |  |
| --- | --- | --- |
| **Rol/Nombre** | **Contacto** | **Expectativas** |
| Patrocinador |  | Garantizar que la arquitectura cumpla con los objetivos estratégicos y de negocio. |
| Equipo de Desarrollo |  | Obtener una arquitectura clara y bien documentada que facilite la implementación. |
| Arquitecto de Software |  | Definir y validar decisiones arquitectónicas en función de los requisitos del sistema. |
| Usuarios Finales |  | Disponer de un sistema intuitivo, eficiente y que cumpla con sus necesidades. |
| Equipo de Calidad |  | Asegurar que la arquitectura cumpla con los estándares de calidad y normativas aplicables. |
| Seguridad TI |  | Verificar que la arquitectura contemple medidas adecuadas de seguridad de la información. |
| Equipo de Mantenimiento |  | Contar con una arquitectura que permita una fácil evolución y corrección de errores. |

* 1. **Restricciones de arquitectura (Architecture Constraints)**

**Contenido**

Las limitaciones de arquitectura son limitaciones necesarias que influyen y definen las decisiones tomadas por los diseñadores y desarrolladores de software con respecto a los aspectos técnicos y operativos de sus proyectos. Estas limitaciones pueden establecerse debido a causas organizativas, políticas o tecnológicas, y, en ciertos escenarios, pueden afectar a toda la compañía.

**Motivación**

Los arquitectos deben ser conscientes de qué aspectos de su diseño están abiertos a la creatividad y, por el contrario, donde la adherencia a directrices o regulaciones estrictas es obligatoria. Siempre considere estas limitaciones, y pueden, en ciertas condiciones, ser flexibles dependiendo de los requerimientos específicos del proyecto. Estas restricciones siempre deben ser consideradas y en algunos casos, pueden ser negociables según las necesidades del proyecto.

**Tipos de Restricciones**

Las restricciones pueden clasificarse en diferentes categorías según su naturaleza:

**5.1. Restricciones Técnicas**

|  |  |
| --- | --- |
| **Restricción** | **Descripción** |
| Lenguajes de programación | Solo se permite el uso de Python para el desarrollo backend. |
| Frameworks | Se debe utilizar Spring Boot para aplicaciones backend y React para frontend. |
| Base de datos | Se requiere el uso de PostgreSQL como sistema de gestión de bases de datos. |
| Infraestructura | La solución debe ser desplegada en la nube utilizando AWS. |

**5.2. Restricciones Organizacionales**

|  |  |
| --- | --- |
| **Restricción** | **Descripción** |
| Metodología de desarrollo | El desarrollo debe seguir la metodología ágil Scrum con iteraciones de dos semanas. |
| Normativas internas | Se deben seguir las políticas de seguridad de información establecidas por la organización. |
| Control de versiones | Uso obligatorio de Git con repositorios alojados en GitHub y Gira |

**5.3. Restricciones Políticas y Regulatorias**

|  |  |
| --- | --- |
| **Restricción** | **Descripción** |
| Cumplimiento normativo | La aplicación debe cumplir con el RGPD (Reglamento General de Protección de Datos). |
| Seguridad de la información | Se deben implementar medidas de seguridad según ISO 27001. |
| Accesibilidad | La interfaz debe cumplir con los estándares de accesibilidad WCAG 2.1. |

**5.4. Convenciones y Estándares**

|  |  |
| --- | --- |
| **Restricción** | **Descripción** |
| Estándares de codificación | Se deben seguir las guías de estilo de Google para Java y PEP 8 para Python. |
| Documentación | Toda documentación debe realizarse en Markdown y almacenarse en Confluence. |
| Nomenclatura | Los nombres de las variables y métodos deben seguir la convención camelCase. |

* 1. **Contexto y Alcance**

**Contenido**

El contexto y el alcance delimitan el sistema ERP de logística de negocios, definiendo sus límites y especificando sus interfaces externas con sistemas vecinos y usuarios. Se diferenciarán los aspectos de contexto de negocio (entradas y salidas específicas del dominio) y el contexto técnico (canales, protocolos y hardware utilizados).

**Motivación**

Las interfaces de dominio y técnicas con los socios de comunicación son aspectos críticos del sistema. Comprenderlas completamente es esencial para garantizar una correcta integración y operatividad del ERP.

**Contexto de negocio**

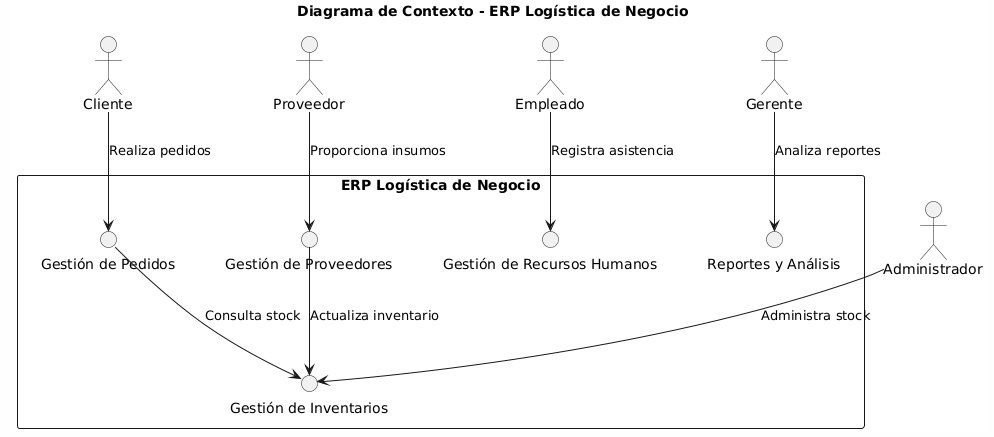
El ERP de logística de negocios interactúa con múltiples actores y sistemas externos. A continuación, se detallan los principales socios de comunicación y sus respectivas interfaces de negocio:

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor / Sistema Externo** | **Descripción** | **Entradas** | **Salidas** |
| Clientes | Empresas que requieren servicios logísticos. | Solicitudes de servicio, datos de carga y destino. | Confirmaciones, estado del servicio, facturación. |
| Proveedores | Empresas que suministran transporte y almacenamiento. | Pedidos de servicio, detalles de carga. | Confirmaciones, costos de servicio, disponibilidad. |
| Operadores logísticos | Personal encargado de gestionar y monitorear operaciones. | Información de rutas, capacidad de carga. | Instrucciones de operación, reportes de estado. |
| Autoridades reguladoras | Entidades gubernamentales que supervisan el cumplimiento de normativas. | Requisitos de cumplimiento, inspecciones. | Reportes, documentación legal. |
| Sistema de Contabilidad | Software externo para la gestión financiera. | Facturación, datos de pagos. | Estados financieros, registros contables. |

**Contexto técnico**

El ERP de logística de negocios interactúa con otros sistemas mediante diversos canales y protocolos. Se detallan a continuación:

|  |  |  |
| --- | --- | --- |
| **Canal / Protocolo** | **Descripción** | **Uso** |
| API REST | Interfaz de comunicación con clientes, proveedores y otros sistemas. | Intercambio de datos en tiempo real. |
| Base de Datos SQL | Almacenamiento y gestión de la información del ERP. | Registro de operaciones, clientes, facturación. |
| Servidor Web | Plataforma para acceso a la aplicación ERP. | Acceso a través de navegadores. |
| Mensajería (MQTT, WebSockets) | Canales de comunicación en tiempo real. | Notificaciones en vivo y actualizaciones de estado. |



* 1. **Contexto de Negocio - ERP Logística de Negocio**

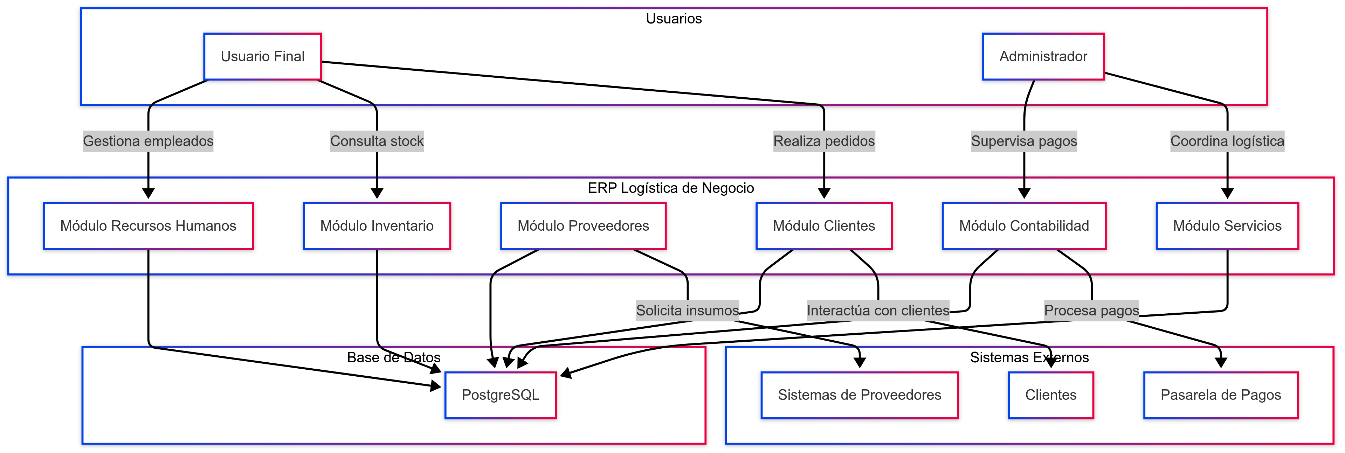
|  |  |  |  |
| --- | --- | --- | --- |
| **Comunicación** | **Partner Externo** | **Entrada al Sistema** | **Salida del Sistema** |
| Gestión de Pedidos | Cliente | Solicitud de pedido, datos de facturación | Confirmación de pedido, factura, estado de entrega |
| Gestión de Inventarios | Proveedor | Órdenes de compra, información de insumos | Confirmación de recepción, actualización de stock |
| Gestión de Proveedores | Administrador | Registro de proveedores, términos de contrato | Lista de proveedores aprobados, estado de contratos |
| Recursos Humanos | Empleado | Registro de asistencia, solicitud de permisos | Aprobación/rechazo de solicitud, historial laboral |
| Reportes y Análisis | Gerente | Parámetros de consulta | Reportes de desempeño, análisis de costos |

**Explicación de interfaces externas**

* **Clientes** interactúan con el sistema mediante formularios de pedidos en la web o aplicación móvil.
* **Proveedores** pueden recibir órdenes de compra y actualizar información de inventario a través de API REST o EDI.
* **Administradores** utilizan un panel web para registrar y gestionar proveedores.
* **Empleados** acceden a su información laboral mediante una aplicación interna.
* **Gerentes** generan reportes con herramientas de Inteligencia de negocio integradas en el ERP.
  1. **Contexto técnico (Technical Context)**

### ****Diagrama UML de Despliegue****

Para representar las interfaces técnicas del sistema, usaremos un diagrama de despliegue en que muestra los canales de comunicación con los sistemas vecinos:



### ****Explicación de Interfaces Técnicas****

1. **Cliente Web/Móvil**
   * **Canal:** HTTPS
   * **Protocolo:** REST API con JSON
   * **Uso:** Permite a los clientes gestionar pedidos, consultar estados y facturas a través de la web o aplicación móvil.
2. **Servidor de Aplicaciones**
   * **Canal:** HTTPS
   * **Protocolo:** REST API
   * **Uso:** Procesa las solicitudes de clientes y administra la lógica de negocio del ERP.
3. **Base de Datos**
   * **Canal:** JDBC
   * **Protocolo:** PostgreSQL
   * **Uso:** Almacena datos transaccionales de pedidos, clientes, inventarios y reportes.
4. **Sistema de Facturación**
   * **Canal:** HTTP
   * **Protocolo:** REST API con JSON
   * **Uso:** Generación y consulta de facturas electrónicas.
5. **Sistema de Inventario**
   * **Canal:** AMQP
   * **Protocolo:** RabbitMQ (Mensajería Asíncrona)
   * **Uso:** Comunicación eficiente de actualizaciones de inventario entre el ERP y almacenes.

### ****Relación Entrada/Salida con los Canales****

|  |  |  |
| --- | --- | --- |
|  | **Entrada (Input)** | **Salida (Output)** |
| HTTPS | Solicitudes de clientes, registros | Confirmación de pedidos, datos de seguimiento |
| REST API | Datos de pedidos, facturas | Respuestas JSON con estados de operación |
| JDBC | Consultas SQL | Datos transaccionales almacenados |
| AMQP | Eventos de cambios de stock | Actualización de inventarios en almacenes |

**Estrategia de Solución (Solution Strategy)**

**Decisiones Fundamentales y Estrategias de Solución**

**Decisiones Tecnológicas**

**Para el desarrollo del ERP Logístico, se han tomado las siguientes decisiones tecnológicas:**

* **Lenguaje de programación:** Uso de Python para el backend.
* **Frameworks:**
  + **Backend:** Django para garantizar escalabilidad y modularidad.
  + **Frontend:** React para una interfaz rápida e interactiva.
* **Base de datos:** PostgreSQL, por su capacidad de manejo de datos estructurados y escalabilidad.
* **Infraestructura:** Despliegue en AWS, aprovechando servicios en la nube para garantizar disponibilidad y rendimiento.

**Patrón Arquitectónico**

El ERP seguirá una arquitectura basada en microservicios, permitiendo escalabilidad y facilidad de mantenimiento. Cada módulo principal (Recursos Humanos, Inventario, Proveedores, Clientes, Contabilidad, Servicios) será un servicio independiente que se comunicará mediante API REST.

**Objetivos de Calidad**

Para garantizar la calidad del ERP Logístico, se establecen los siguientes objetivos:

1. Rendimiento: Tiempo de respuesta menor a 2 segundos en operaciones clave.
2. Seguridad: Implementación de cifrado de datos, autenticación basada en roles y auditoría de registros.
3. Mantenibilidad: Arquitectura modular y documentada para facilitar la extensión del sistema.
4. Usabilidad: Interfaz intuitiva, compatible con dispositivos móviles.
5. **Interoperabilidad: Integración con otros sistemas a través de API REST.**

**Decisiones Organizativas**

* **Metodología de desarrollo**: Uso de Scrum, con iteraciones de dos semanas.
* **Control de versiones:** Uso de Git con repositorios en GitHub.
* **Normativas:** Cumplimiento con RGPD e implementación de seguridad basada en ISO 27001.

**Vista de Bloques de Construcción (Building Block View)**

**Estructura Estática del Sistema**

**El ERP Logístico se descompone en los siguientes bloques:**

**Nivel 1:** Descripción General del Sistema

El sistema se organiza en módulos independientes:

* **Recursos Humanos:** Gestión de empleados, roles y asistencia.
* **Inventario**: Control de existencias, asignación de recursos.
* **Proveedores**: Registro y evaluación de proveedores.
* **Clientes**: Administración de contratos y facturación.
* **Contabilidad**: Gestión financiera y presupuestaria.
* **Servicios**: Programación y seguimiento de operaciones logísticas.

**Nivel 2: Interna de los Módulos**

Cada módulo se descompone en subcomponentes. Por ejemplo, el módulo de Inventario se divide en:

* Gestión de productos
* Control de stock
* Alertas de reabastecimiento
* Historial de movimientos

**Nivel 3: Componentes Internos**

**Vista de Tiempo de Ejecución (Runtime View)**

**Escenarios Clave**

**Escenario 1: Creación de un Pedido**

1. El cliente realiza una solicitud a través de la web/móvil.
2. El sistema verifica disponibilidad en el inventario.
3. Se genera la orden y se notifica al proveedor.
4. Se actualiza el stock y se genera una factura.
5. El cliente recibe confirmación.

**Escenario 2: Control de Inventario**

1. Un empleado registra la entrada de nuevos productos.
2. El sistema actualiza el stock y envía alertas si es necesario.
3. Se generan reportes automáticos de movimientos.

**Vista de Despliegue (Deployment View)**

**Infraestructura del Sistema**

**El ERP Logístico se desplegará en la nube utilizando AWS, con la siguiente configuración:**

* **Frontend:** Servido desde AWS S3 + CloudFront para distribución de contenido rápido y eficiente.
* **Backend:** Desplegado en AWS EC2 con balanceo de carga mediante AWS Elastic Load Balancer para escalabilidad.
* **Base de datos:** PostgreSQL en AWS RDS, asegurando redundancia y alta disponibilidad.
* **Mensajería:** RabbitMQ para eventos en tiempo real, facilitando la comunicación entre microservicios.
* **Almacenamiento de archivos:** Uso de AWS S3 para almacenamiento seguro y accesible.

**Topología de Despliegue**

**El sistema está diseñado para operar en un entorno distribuido, con la siguiente infraestructura:**

1. **Zona de Disponibilidad Múltiple:** Uso de múltiples zonas en AWS para tolerancia a fallos.
2. **Red Privada Virtual (VPC):** Separación de componentes internos del acceso público.
3. **Capa de Seguridad:** Uso de grupos de seguridad y VPN para acceso restringido a recursos sensibles.

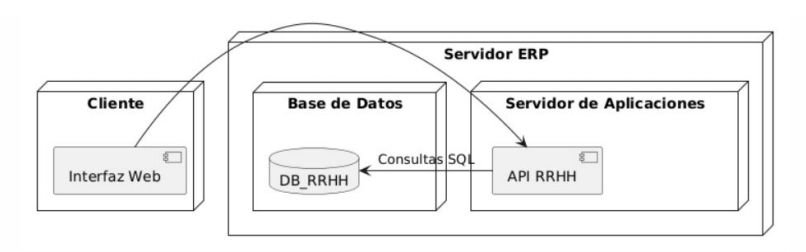
**Ambientes de Ejecución**

**El sistema operará en tres entornos diferenciados para optimizar el desarrollo y despliegue:**

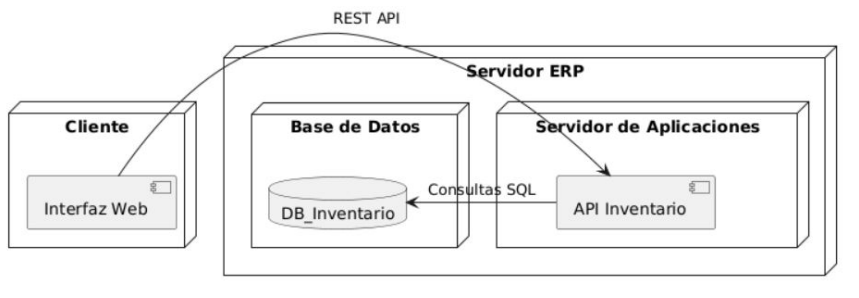
1. **Desarrollo:** S ervidores locales para pruebas iniciales y depuración.
2. **Pruebas:** Infraestructura en la nube replicando el entorno de producción.
3. **Producción:** Configuración escalable con monitoreo en tiempo real mediante AWS CloudWatch.

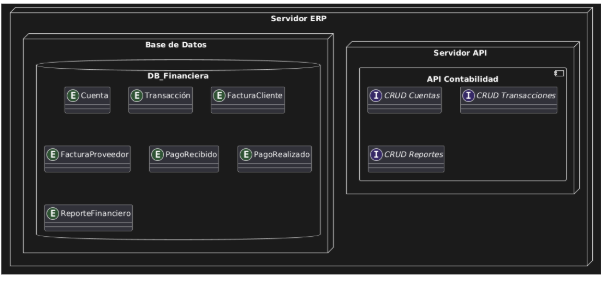
**Diagramas de Despliegue**

**RRHH**

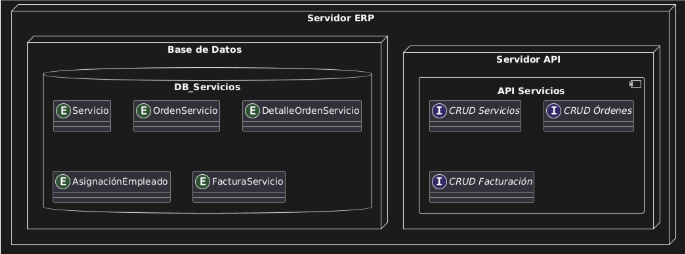


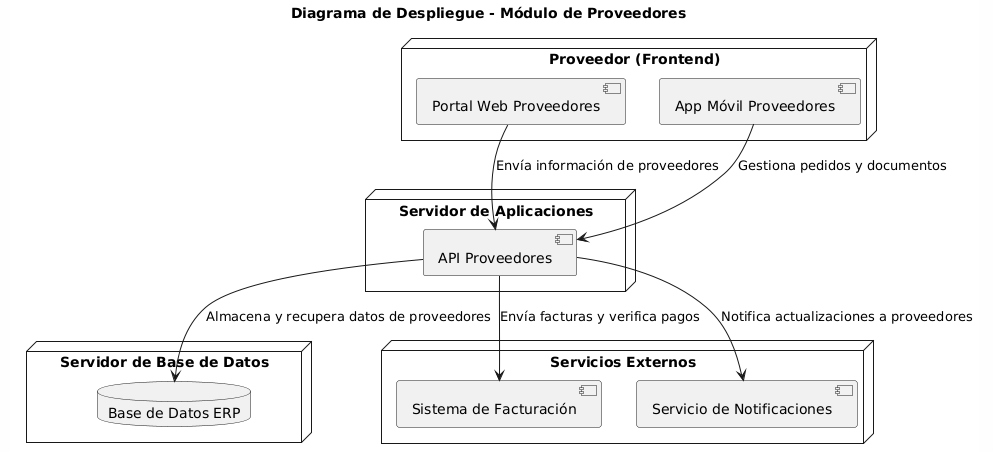
**inventario**

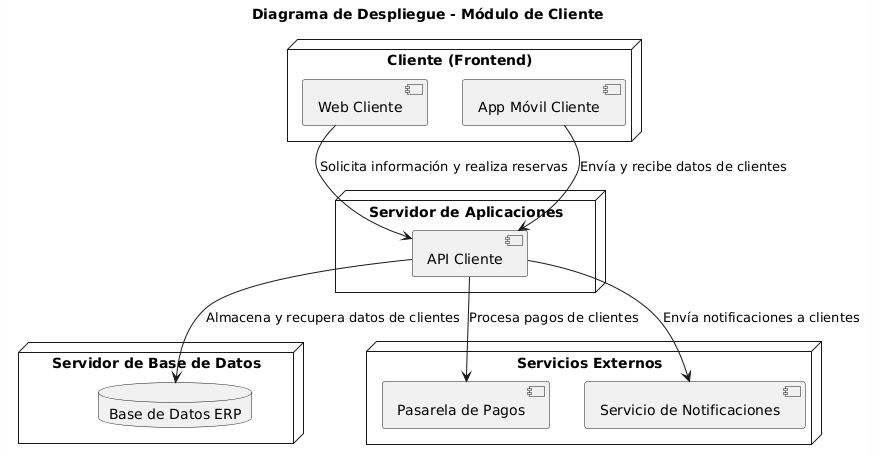


Contabilidad

Servicios



Cliente

Proveedores

* 1. **Patrones de arquitectura**

### ****Patrones arquitectónicos aplicados:****

**MVC (Modelo-Vista-Controlador):**  
Separación de responsabilidades entre la lógica de negocio (Modelo), la interfaz de usuario (Vista) y el flujo de control (Controlador). Este patrón es nativo en Django, el framework elegido.

**Arquitectura en capas (Layered Architecture):**  
Se divide el sistema en capas: presentación, lógica de negocio y acceso a datos, lo cual favorece la organización, reutilización y mantenibilidad del código.

**Cliente-Servidor:**  
La comunicación ocurre entre clientes (navegadores web o móviles) y el servidor backend en Django que gestiona lógica y persistencia de datos.

### ****Conceptos clave usados:****

**Componentización:**  
El sistema se divide en componentes independientes (usuarios, productos, pedidos, etc.), lo cual permite desarrollar y probar por separado.

**APIs RESTful:**  
Comunicación entre frontend y backend mediante servicios REST para obtener o enviar información como productos, pedidos o usuarios.

**Autenticación y autorización:**  
Uso del sistema de autenticación de Django para registrar e iniciar sesión, así como para asignar roles y permisos diferenciados (usuarios vs. administradores).

**Responsividad:**  
Interfaz diseñada para adaptarse correctamente a distintos dispositivos (móviles, tablets, escritorio), usando CSS y JavaScript.

**Seguridad:**  
Uso de hash para contraseñas, protección contra CSRF y validaciones de entrada para mitigar riesgos comunes en aplicaciones web.

**10. Requisitos de calidad**

### ****Seguridad****

**Escenario:**  
Cuando un usuario inicia sesión, el sistema debe verificar sus credenciales cifradas y proteger contra ataques de inyección, CSRF y XSS.  
**Objetivo de calidad:** Prevención de accesos no autorizados y ataques comunes.  
**Medidas:**

* Uso del sistema de autenticación de Django.
* Cifrado de contraseñas con hashing (PBKDF2 por defecto).
* Middleware CSRF activo.

### ****Rendimiento****

**Escenario:**  
Cuando un usuario agrega un producto al carrito o realiza un pedido, la operación debe completarse en menos de 2 segundos bajo carga normal.  
**Objetivo de calidad:** Tiempo de respuesta rápido para una experiencia fluida.  
**Medidas:**

* Indexación de base de datos.
* Carga asincrónica en frontend.
* Optimización de consultas.

### ****Usabilidad****

**Escenario:**  
Un usuario debe poder buscar, filtrar y comprar productos fácilmente desde un dispositivo móvil.  
**Objetivo de calidad:** Interfaz intuitiva, accesible y responsiva.  
**Medidas:**

* Uso de diseño responsivo (CSS adaptable).
* Navegación clara y categorización por productos.
* Etiquetas y formularios comprensibles.

### ****Mantenibilidad****

**Escenario:**  
Cuando se deba modificar el módulo de pedidos, otros módulos no deben verse afectados.  
**Objetivo de calidad:** Bajo acoplamiento y alta cohesión.  
**Medidas:**

* Separación por componentes (apps en Django).
* Uso de principios SOLID.
* Pruebas unitarias por componente.

**11. Riesgos y deuda técnica**

### ****Riesgos del sistema****

|  |  |  |
| --- | --- | --- |
| **Riesgo** | **Impacto** | **Mitigación** |
| Retrasos en el desarrollo por falta de experiencia con Django | Alto | Capacitación previa y revisión de avances por parte del equipo |
| Escalabilidad limitada si aumenta el número de usuarios | Medio | Diseñar desde el inicio con escalabilidad en mente (uso de capas, optimización de base de datos) |
| Integración deficiente entre módulos (productos, usuarios, pedidos) | Medio | Realizar pruebas de integración frecuentes |
| Pérdida de datos por errores en formularios o caídas | Alto | Validaciones sólidas, backups automáticos y manejo de errores |

### ****Deuda técnica estimada****

|  |  |  |
| --- | --- | --- |
| **Elemento** | **Descripción** | **Posible solución futura** |
| Código duplicado en plantillas HTML | Fragmentos de código repetido en vistas similares | Usar herencia de plantillas y componentes reutilizables |
| Pruebas unitarias limitadas | No se implementaron pruebas completas durante desarrollo inicial | Añadir pruebas usando pytest o unittest progresivamente |
| Validación mínima del lado cliente | Los formularios validan principalmente en backend | Incorporar validación con JavaScript en el frontend |

1. **Glosario**

|  |  |
| --- | --- |
| **Término** | **Definición** |
| **Django** | Framework web de alto nivel en Python que fomenta el desarrollo rápido y limpio. |
| **MVC** | P*atrón de ar*quitectura Modelo-Vista-Controlador usado para estructurar el sistema. |
| **REST API** | Interfaz que permite la comunicación entre frontend y backend mediante peticiones HTTP. |
| **PostgreSQL** | Sistema de gestión de bases de datos relacional usado en el proyecto. |
| **Carrito de compras** | Funcionalidad que permite a los usuarios almacenar productos temporalmente antes de comprarlos. |
| **CRUD** | Operaciones básicas: Crear, Leer, Actualizar y Eliminar datos. |
| **Hashing** | Técnica de cifrado utilizada para proteger contraseñas. |