| DUOC UC - ESCUELA DE INFORMÁTICA Y TELECOMUNICACIONES |
| --- |
| Especificación de Requisitos de Software |
| Proyecto: ComparaYa |
|  |
| **Revisión*: [01.0]*** |
| **[11/08/2025]** |

| Especificación de Requisitos según estándar de IEEE 830. |
| --- |

**Contenido**

[**1. Introducción 5**](#_heading=h.1fob9te)

[1.1. Propósito 5](#_heading=h.3znysh7)

[1.2. Ámbito del Sistema 6](#_heading=h.2et92p0)

[Funcionalidades principales 6](#_heading=h.i4o6iartelp1)

[**Usuarios del sistema 7**](#_heading=h.waccxfrx3szc)

[**Límites del sistema 7**](#_heading=h.ko1zakcpx2op)

[1.3. Definiciones, Acrónimos y Abreviaturas 7](#_heading=h.tyjcwt)

[1.4. Referencias 9](#_heading=h.3dy6vkm)

[1.5. Visión General del Documento 9](#_heading=h.1t3h5sf)

[**2. Descripción General 10**](#_heading=h.4d34og8)

[**Perspectiva del Producto 10**](#_heading=h.zc6j6l6e5j0z)

[**Funciones del Sistema 10**](#_heading=h.mmh5hdmzuzig)

[**Características de los Usuarios 11**](#_heading=h.y5szjhd0lemy)

[**Restricciones 11**](#_heading=h.o5211h2yaphq)

[**Supuestos y Dependencias 11**](#_heading=h.s1nhx33ctvz5)

[2.1. Perspectiva del Producto 11](#_heading=h.2s8eyo1)

[**Contexto 11**](#_heading=h.dzx1n9wv84is)

[**Interfaces del Producto 12**](#_heading=h.bp0z7e98g1ga)

[**Relación con Otros Sistemas 12**](#_heading=h.ccmcy8lh3b1o)

[**Beneficios Esperados 12**](#_heading=h.lzqa016e1nrp)

[2.2. Funciones del Producto 13](#_heading=h.17dp8vu)

[2.3. Características de los Usuarios 14](#_heading=h.3rdcrjn)

[2.4. Restricciones 15](#_heading=h.26in1rg)

[2.5. Suposiciones y Dependencias 16](#_heading=h.lnxbz9)

[**Suposiciones 16**](#_heading=h.lhr7yn1a3yot)

[**Dependencias 16**](#_heading=h.rmqb5w4s3vzz)

[2.6. Requisitos Futuros 17](#_heading=h.35nkun2)

[**3. Requisitos Específicos 18**](#_heading=h.1ksv4uv)

[3.1 Requisitos comunes de las interfaces 18](#_heading=h.44sinio)

[3.1.1 Interfaces de usuario 18](#_heading=h.2jxsxqh)

[3.1.2 Interfaces de hardware 18](#_heading=h.z337ya)

[3.1.3 Interfaces de software 19](#_heading=h.3j2qqm3)

[3.1.4 Interfaces de comunicación 19](#_heading=h.1y810tw)

[3.2 Requisitos funcionales 20](#_heading=h.zer0i04fkequ)

[3.3 Requisitos no funcionales 21](#_heading=h.h9vxvesxq25d)

[3.3.1 Requisitos de rendimiento 21](#_heading=h.3gy6ckymnrbh)

[3.3.2 Seguridad 21](#_heading=h.2bn6wsx)

[3.3.3 Fiabilidad 21](#_heading=h.qsh70q)

[3.3.4 Disponibilidad 21](#_heading=h.3as4poj)

[3.3.5 Mantenibilidad 22](#_heading=h.1pxezwc)

[3.3.6 Portabilidad 22](#_heading=h.49x2ik5)

[3.4 Otros Requisitos 22](#_heading=h.2p2csry)

**Ficha del documento**

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
| 05/09/2025 | 1.0 | Gustavo Cartes | Versión inicial |

Documento validado por las partes en fecha:

| Por el cliente |  | Por la empresa suministradora |
| --- | --- | --- |
| [Firma] |  | [Firma] |
| Sr./Sra. |  | Sr./Sra. |

# Introducción

El presente documento corresponde a la **Especificación de Requerimientos de Software (ERS)** del proyecto **ComparaYa**, una plataforma web orientada a la comparación y análisis de precios de productos agrícolas y de consumo en Chile.

El sistema surge de la necesidad de contar con una herramienta confiable que permita a los usuarios visualizar y comparar, de manera clara y accesible, la información histórica y actual de precios publicada por organismos oficiales. De esta forma, se busca mejorar la toma de decisiones tanto a nivel de consumidores como de mayoristas, contribuyendo a la transparencia del mercado.

El proyecto **ComparaYa** contempla dos funcionalidades principales:

1. **Dashboards interactivos**: permiten analizar la evolución de precios de distintos productos a través de gráficos dinámicos y filtros por variables como región, unidad de medida y período de tiempo.
2. **Comparador de productos**: facilita la selección de dos o más productos para visualizar sus diferencias de precios, tendencias y disponibilidad, entregando al usuario información práctica para la compra y planificación.

Este documento describe de manera detallada los objetivos, alcance, requerimientos funcionales y no funcionales del sistema, constituyéndose como base para el desarrollo, implementación y posterior validación del proyecto.

## Propósito

El propósito del sistema **ComparaYa** es desarrollar una plataforma digital que centralice, organice y presente de forma clara la información de precios de productos agrícolas y de consumo en Chile.

La aplicación está orientada a:

* **Consumidores**: entregarles una herramienta sencilla para comparar precios y tendencias, apoyando decisiones de compra más informadas.
* **Mayoristas y comerciantes**: proporcionar dashboards con indicadores y análisis que faciliten la planificación de compra y venta.
* **Investigadores y organismos públicos**: brindar acceso a datos históricos estructurados que puedan servir para estudios de mercado y políticas públicas.

Con ello, se busca **aumentar la transparencia en la información de precios, facilitar el acceso a datos confiables y apoyar la toma de decisiones económicas** en distintos niveles de usuario.

## Ámbito del Sistema

El sistema **ComparaYa** será una aplicación web responsiva que permitirá a los usuarios acceder a información actualizada y confiable sobre precios de productos agrícolas y de consumo. El sistema se conectará a bases de datos generadas a partir de fuentes oficiales (por ejemplo, ODEPA), procesará los archivos de datos mediante procesos ETL y presentará la información en interfaces interactivas y fáciles de usar.

### Funcionalidades principales

1. **Dashboards de productos**
   * Visualización de precios históricos y actuales mediante gráficos interactivos.
   * Filtros por producto, región, unidad de medida y período de tiempo.
   * Opciones de agregación (promedio, máximo, mínimo).
2. **Comparador de productos**
   * Selección de dos o más productos para su comparación.
   * Visualización de tendencias, diferencias de precio y comportamiento histórico.
   * Presentación clara de resultados a través de tablas y gráficos.
3. **Gestión de datos**
   * Procesamiento automático de archivos CSV provenientes de las fuentes oficiales.
   * Limpieza, validación y almacenamiento de datos en la base de datos central.
   * Generación de reportes y métricas de uso.

### Usuarios del sistema

* **Consumidores finales**: accederán al comparador para tomar decisiones de compra.
* **Comerciantes y mayoristas**: utilizarán dashboards para planificar precios y volúmenes de venta.
* **Investigadores/organismos públicos**: podrán consultar series históricas de precios y obtener análisis agregados.

### Límites del sistema

* El sistema se limita a la información publicada por fuentes oficiales de datos (no genera información propia).
* No realiza transacciones de compra o venta, solo provee información para apoyar decisiones.
* La predicción de precios (forecasting) se considera un módulo complementario y no es parte del alcance inicial obligatorio.

## Definiciones, Acrónimos y Abreviaturas

* **ComparaYa:** Nombre del sistema web que permite visualizar dashboards de precios y realizar comparaciones de productos agrícolas y de consumo.
* **Dashboard:** Interfaz gráfica que muestra indicadores clave, gráficos y métricas de manera resumida y visual.
* **Comparador:** Funcionalidad del sistema que permite contrastar precios y tendencias entre dos o más productos.
* **Producto:** Bien agrícola o de consumo registrado en la base de datos con información de precios asociados.
* **Unidad de medida:** Forma en que se expresa el precio del producto (ejemplo: kilo, litro, caja de 18 kilos, etc.).
* **Región:** División geográfica utilizada para la segmentación de los datos en Chile.
* **CSV (Comma-Separated Values):** Formato de archivo de texto plano utilizado para la carga de datos en el sistema.
* **ETL (Extract, Transform, Load):** Proceso de extracción, transformación y carga de datos desde archivos fuente hacia la base de datos del sistema.
* **API (Application Programming Interface):** Conjunto de endpoints que permiten acceder a la información procesada desde aplicaciones externas o el frontend del sistema.
* **Frontend:** Parte del sistema visible para el usuario, desarrollada en React.
* **Backend:** Parte lógica y de gestión de datos del sistema, desarrollada en Django REST Framework.
* **PostgreSQL:** Sistema de gestión de bases de datos relacional utilizado para almacenar la información del sistema.
* **Forecasting:** Módulo complementario para la predicción de precios futuros en base a series históricas (fuera del alcance inicial).

## 1.4. Referencias

* **IEEE 830-1998**. *Recommended Practice for Software Requirements Specifications*. Estándar utilizado como guía para la elaboración del presente documento.
* **ODEPA (Oficina de Estudios y Políticas Agrarias, Chile)**. Sitio oficial con reportes y archivos de precios agrícolas y de consumo. Disponible en: https://www.odepa.gob.cl
* **PostgreSQL Global Development Group**. *PostgreSQL Documentation*. Sistema de gestión de bases de datos utilizado en el proyecto. Disponible en:<https://www.postgresql.org/docs>
* **Django Software Foundation**. *Django Documentation*. Framework backend utilizado en el desarrollo del sistema. Disponible en: https://docs.djangoproject.com
* **Django REST Framework**. *DRF Documentation*. Librería para construcción de APIs. Disponible en: https://www.django-rest-framework.org
* **ReactJS**. *React Documentation*. Librería de JavaScript utilizada para el desarrollo del frontend. Disponible en: https://react.dev
* **Tailwind CSS**. *Tailwind Documentation*. Framework de estilos CSS utilizado en la interfaz del sistema. Disponible en:<https://tailwindcss.com>

## 1.5. Visión General del Documento

El presente documento tiene como objetivo describir de manera estructurada y formal los requerimientos del sistema **ComparaYa**. Para ello, se organiza en secciones que permiten comprender desde el contexto general hasta los detalles específicos del software a desarrollar.

En particular, este documento incluye:

1. **Introducción**: Presenta el propósito del sistema, su ámbito de aplicación, las definiciones y las referencias utilizadas.
2. **Descripción general**: Proporciona una visión de alto nivel del sistema, su contexto, usuarios, funciones principales, restricciones y supuestos.
3. **Requerimientos específicos**: Define los requerimientos funcionales y no funcionales del sistema de manera detallada, describiendo qué debe hacer el software y bajo qué condiciones debe operar.
4. **Anexos**: Incluyen información complementaria relevante, como diagramas, ejemplos de datos, estructuras de API, entre otros.

Este documento servirá como base para el diseño, desarrollo, pruebas e implementación de **ComparaYa**, asegurando que todas las partes interesadas tengan una comprensión clara y compartida de las funcionalidades y objetivos del sistema.

# 2. Descripción General

El sistema **ComparaYa** es una aplicación web diseñada para centralizar, procesar y presentar información de precios de productos agrícolas y de consumo en Chile, con el objetivo de mejorar la transparencia y apoyar la toma de decisiones.

### Perspectiva del Producto

* **Tipo de sistema**: Aplicación web responsiva con arquitectura cliente–servidor.
* **Frontend**: Desarrollado en **React** y estilizado con **Tailwind CSS**, enfocado en la usabilidad y visualización de datos mediante dashboards y comparadores.
* **Backend**: Desarrollado en **Django REST Framework**, encargado de la gestión de datos, lógica de negocio y exposición de APIs.
* **Base de Datos**: **PostgreSQL**, utilizada para almacenar datos históricos y actuales de precios, junto con metadatos de los procesos ETL.
* **Procesos ETL**: Implementados en **Python**, responsables de la extracción, limpieza y carga de datos provenientes de fuentes oficiales como ODEPA.

### Funciones del Sistema

1. Dashboards de visualización de precios con filtros por producto, región, unidad y período.
2. Comparador de productos que muestra diferencias de precio y tendencias históricas.
3. Procesamiento automático de archivos CSV (ETL) para mantener la base de datos actualizada.
4. Exposición de una API que permite el acceso estructurado a los datos para el frontend y posibles integraciones futuras.

### Características de los Usuarios

* **Consumidores finales**: buscan comparar precios y tomar decisiones informadas de compra.
* **Comerciantes y mayoristas**: requieren dashboards para planificar precios y ventas.
* **Investigadores y entidades públicas**: consultan datos históricos y tendencias agregadas.

### Restricciones

* Dependencia de los datos entregados por fuentes oficiales (ej. ODEPA).
* El sistema no realiza transacciones de compra o venta, solo provee información.
* El módulo de predicción de precios (forecasting) es opcional y no forma parte del alcance inicial obligatorio.

### Supuestos y Dependencias

* Los datos estarán disponibles de forma periódica y en formato estándar (CSV).
* Los usuarios tendrán acceso a internet y a un navegador actualizado para utilizar la plataforma.
* La infraestructura del servidor soportará la carga esperada de usuarios concurrentes.

## 2.1. Perspectiva del Producto

El sistema **ComparaYa** se concibe como una aplicación web de análisis y comparación de precios agrícolas y de consumo, cuyo propósito principal es transformar información cruda proveniente de fuentes oficiales en herramientas interactivas y comprensibles para distintos tipos de usuarios.

### Contexto

* El sistema funcionará bajo una arquitectura **cliente-servidor**, donde el frontend se conectará a un backend mediante servicios REST.
* Se alimentará de datos oficiales proporcionados por la **ODEPA** en formato CSV, los cuales serán procesados mediante rutinas ETL (Extract, Transform, Load).
* La información procesada se almacenará en una base de datos **PostgreSQL**, permitiendo consultas optimizadas y consistentes.

### Interfaces del Producto

* **Frontend (React + Tailwind CSS):** interfaz gráfica que presentará dashboards interactivos y el módulo de comparador.
* **Backend (Django REST Framework):** encargado de la lógica de negocio, gestión de datos y exposición de la API.
* **Base de Datos (PostgreSQL):** repositorio estructurado de los datos históricos y actuales de precios.
* **ETL (Python):** conjunto de procesos automatizados que realizan la descarga, limpieza, transformación y carga de los datos desde las fuentes originales.

### Relación con Otros Sistemas

* El sistema se integra únicamente con fuentes externas de datos (ODEPA u organismos similares).
* En versiones futuras, se contempla la posibilidad de abrir la API para consumo de terceros interesados en información de precios.

### Beneficios Esperados

* Transparencia en la información de precios agrícolas y de consumo.
* Herramienta de apoyo para consumidores, comerciantes y organismos públicos.
* Centralización y estandarización de datos dispersos en un sistema confiable y fácil de usar.

## 2.2. Funciones del Producto

El sistema **ComparaYa** proporcionará las siguientes funciones principales:

1. **Dashboards de Precios**
   * Visualización de series históricas y actuales de precios.
   * Aplicación de filtros por producto, región, unidad de medida y período de tiempo.
   * Agregación de datos (promedio, máximo, mínimo) para un mejor análisis.
   * Representación en gráficos interactivos (líneas, barras, etc.) y tablas.
2. **Comparador de Productos**
   * Selección de dos o más productos para su comparación.
   * Visualización de diferencias de precios en un mismo período o región.
   * Comparación de tendencias históricas mediante gráficos superpuestos.
   * Presentación de resultados en tablas comparativas.
3. **Gestión de Datos (ETL)**
   * Descarga automática de archivos CSV desde fuentes oficiales (ej. ODEPA).
   * Limpieza y transformación de los datos para normalizar unidades, formatos y valores.
   * Carga de la información a la base de datos central (PostgreSQL).
   * Registro de metadatos de cada carga (fecha, estado, número de registros procesados).
4. **Exposición de API**
   * Endpoints para acceder a catálogos de productos, precios, unidades y regiones.
   * Servicios para alimentar el frontend y permitir integraciones futuras.
   * Paginación y filtros en las consultas para mejorar el rendimiento.
5. **Módulo de Administración *(versión futura opcional)***
   * Panel de control para monitorear procesos ETL.
   * Gestión de logs de errores y reportes de carga.
   * Supervisión del estado de la base de datos y consumo de la API.

## 2.3. Características de los Usuarios

| **Perfil** | **Descripción** | **Objetivos principales** | **Nivel técnico** |
| --- | --- | --- | --- |
| Consumidor final | Persona común interesada en comparar precios de productos agrícolas y de consumo. | - Consultar precios actuales.  - Comparar productos entre distintas regiones o formatos. | Básico (uso de navegador). |
| Comerciante/Mayorista | Usuario que compra o vende productos en volumen y requiere datos de mercado. | - Analizar tendencias históricas.  - Planificar precios y volúmenes de compra/venta. | Intermedio (manejo de dashboards). |
| Investigador/Académico | Personas vinculadas a estudios de mercado, economía o agricultura. | - Obtener series históricas completas.  - Descargar datos para análisis externos. | Intermedio-Avanzado (uso de datos). |
| Entidad pública | Organismos gubernamentales o municipales que necesitan datos para políticas públicas. | - Acceder a datos confiables y centralizados.  - Evaluar tendencias macroeconómicas. | Intermedio. |
| Administrador del sistema *(futuro)* | Responsable técnico del mantenimiento del sistema. | - Supervisar procesos ETL.  - Gestionar logs y rendimiento del sistema. | Avanzado (gestión técnica). |

## 2.4. Restricciones

El desarrollo y funcionamiento del sistema **ComparaYa** estará sujeto a las siguientes limitaciones:

1. **Dependencia de fuentes externas de datos**
   * El sistema se alimenta de archivos publicados por la ODEPA y otras fuentes oficiales.
   * La actualización de la información dependerá de la disponibilidad y formato de dichos archivos.
2. **Formato de los datos**
   * Los datos deben provenir en formato **CSV** con estructura predefinida.
   * Cambios en la estructura de los archivos fuente pueden requerir ajustes en los procesos ETL.
3. **Alcance funcional**
   * El sistema no realiza transacciones de compra o venta, únicamente muestra información.
   * El módulo de predicción de precios (forecasting) se considera un complemento y no está en el alcance inicial obligatorio.
4. **Acceso al sistema**
   * El sistema será accesible únicamente vía navegador web.
   * Requiere conexión a internet estable para el consumo de datos y visualización.
5. **Limitaciones técnicas**
   * La capacidad de procesamiento y almacenamiento estará restringida por los recursos del servidor implementado.
   * La concurrencia de usuarios dependerá de la infraestructura de despliegue configurada.
6. **Seguridad y privacidad**
   * El sistema no gestionará datos sensibles de usuarios, limitándose únicamente a información pública de precios.
   * Cualquier registro adicional de usuarios (ej. administrador) deberá cumplir con estándares básicos de seguridad.

## 2.5. Suposiciones y Dependencias

El desarrollo y operación del sistema **ComparaYa** se fundamenta en los siguientes supuestos y dependencias:

### Suposiciones

1. Los datos de precios estarán disponibles de forma periódica y pública en el sitio web de la **ODEPA** u otras fuentes oficiales.
2. Los archivos proporcionados por dichas fuentes se mantendrán en formato **CSV** o equivalente, con estructuras consistentes.
3. Los usuarios contarán con conexión a internet y un navegador moderno actualizado (ej. Chrome, Firefox, Edge).
4. El volumen de datos históricos será manejable por los procesos ETL definidos en el sistema.
5. El sistema no requiere autenticación de usuarios finales en su primera versión, dado que la información es pública.

### Dependencias

1. **Dependencia de datos externos**: La plataforma depende de la disponibilidad, consistencia y continuidad de los archivos publicados por ODEPA.
2. **Dependencia tecnológica**:  
   * **Backend**: Django REST Framework y Python.
   * **Frontend**: React con Tailwind CSS.
   * **Base de datos**: PostgreSQL.
   * **Procesos ETL**: scripts en Python.
3. **Dependencia de infraestructura**:  
   * Servidor web con capacidad suficiente para procesar datos y soportar la concurrencia esperada.
   * Almacenamiento en disco para datos históricos y archivos temporales.
4. **Dependencia de librerías externas**: El sistema utiliza librerías y frameworks de terceros que deben mantenerse actualizados y compatibles.

## 2.6. Requisitos Futuros

El sistema **ComparaYa** podrá ampliarse en versiones posteriores con las siguientes funcionalidades y mejoras:

1. **Módulo de Predicción de Precios (Forecasting)**
   * Implementación de modelos estadísticos y de machine learning (ARIMA, Prophet, etc.) para estimar precios futuros.
   * Visualización de proyecciones en los dashboards junto con las series históricas.
2. **Exportación de Datos**
   * Posibilidad de descargar series históricas en formatos **CSV, Excel o PDF**.
   * Generación de reportes automáticos para investigadores, comerciantes y entidades públicas.
3. **Notificaciones y Alertas**
   * Envío de alertas personalizadas cuando un producto alcance un precio definido por el usuario.
   * Notificaciones sobre cambios importantes de tendencias.
4. **Autenticación y Perfiles de Usuario**
   * Creación de cuentas para que los usuarios guarden preferencias, productos favoritos o configuraciones de alertas.
   * Distintos niveles de acceso (ej. consumidor, mayorista, administrador).
5. **Panel de Administración Avanzado**
   * Monitoreo en tiempo real de procesos ETL.
   * Gestión de errores y reportes de auditoría.
   * Configuración de integraciones con nuevas fuentes de datos.
6. **Integraciones Externas**
   * API pública para que terceros puedan consumir datos de precios.
   * Posible integración con aplicaciones móviles en futuras versiones.

# 3. Requisitos Específicos

Todos los requisitos que realizará el aplicativo web **ComparaYa** están dirigidos a los usuarios finales y a los administradores del sistema. En esta sección se detallan las diferentes interfaces y funcionalidades esperadas.

## 3.1 Requisitos comunes de las interfaces

Descripción común de entrada y salida de las interfaces.

### 3.1.1 Interfaces de usuario

La interfaz de usuario consistirá en un conjunto de pantallas web con menús, botones, listas desplegables y filtros que permitirán visualizar dashboards interactivos y realizar comparaciones de productos. La información se mostrará mediante gráficos (líneas, barras, etc.) y tablas de datos.

### 3.1.2 Interfaces de hardware

Será necesario que los dispositivos cliente dispongan de:

* Tarjeta de red funcional.
* Acceso a internet.
* Navegador actualizado (desktop, laptop, tablet o móvil).

### 3.1.3 Interfaces de software

El aplicativo web será compatible con los navegadores más utilizados: Google Chrome, Mozilla Firefox, Microsoft Edge y Safari. El backend funcionará sobre **Django REST Framework** y el frontend sobre **React + Tailwind CSS**.

### 3.1.4 Interfaces de comunicación

Los servidores, clientes y aplicativo se comunicarán mediante protocolos estándar de internet (HTTP/HTTPS). El consumo de datos se realizará mediante **API REST** en formato **JSON**.

## 3.2 Requisitos funcionales

En la siguiente tabla se presentan los requisitos que son funcionales del aplicativo web.

| **Número de Requisito** | **Nombre de Requisito** | **Prioridad del Requisito** | **Descripción del Requisito** |
| --- | --- | --- | --- |
| RF 01 | Comparar Precios | Alta / Esencial | El sistema debe permitir comparar precios de dos o más productos, mostrando sus diferencias históricas y actuales. |
| RF 02 | Visualización en Dashboards | Alta / Esencial | El sistema debe mostrar dashboards con series históricas de precios y permitir aplicar filtros por región, unidad y período. |
| RF 03 | Filtrado Avanzado | Alta / Esencial | El sistema debe incluir filtros dinámicos para consultar por producto, región, unidad de medida y rango de fechas. |
| RF 04 | Procesamiento ETL | Alta / Esencial | El sistema debe descargar, limpiar, transformar y cargar automáticamente los datos desde fuentes oficiales (ej. ODEPA). |
| RF 05 | Registro de Metadatos | Media | Cada proceso ETL debe registrar fecha, estado y número de registros procesados. |
| RF 06 | Exposición de API | Alta | El sistema debe exponer endpoints REST para productos, regiones, unidades y series de precios. |
| RF 07 | Paginación y Filtros en API | Media | Los endpoints deben permitir paginación y filtros en las consultas. |
| RF 08 | Respuesta en JSON | Alta | Toda respuesta de la API debe entregarse en formato JSON. |
| RF 09 | Interfaz Responsiva | Alta | El sistema debe ser accesible desde distintos dispositivos (desktop, móvil, tablet). |
| RF 10 | Manejo de Errores | Alta | El sistema debe mostrar mensajes claros en caso de errores o ausencia de datos |

## 3.3 Requisitos no funcionales

En la siguiente tabla se representan los requisitos que son no funcionales del aplicativo web.

| Identificación | Nombre | Descripción | Prioridad |
| --- | --- | --- | --- |
| RNF 01 | Tiempo de respuesta | Las consultas a la API deben responder en menos de 2 segundos bajo carga normal. | Alta |
| RNF 02 | Disponibilidad | El sistema debe estar disponible 24/7 con un tiempo de actividad ≥ 95%. | Alta |
| RNF 03 | Portabilidad | El sistema debe poder ejecutarse en diferentes navegadores y dispositivos. | Alta |
| RNF 04 | Documentación | El sistema debe disponer de documentación técnica clara para su mantenimiento. | Alta |
| RNF 05 | Escalabilidad | El sistema debe permitir la incorporación de nuevos productos o fuentes de datos sin rediseño mayor. | Media |

### 3.3.1 Requisitos de rendimiento

* El sistema debe soportar al menos 100 usuarios concurrentes en su primera versión.
* La carga y procesamiento de datos ETL no debe sobrepasar 5 minutos por archivo CSV bajo condiciones normales.

### 3.3.2 Seguridad

* Uso obligatorio de **HTTPS** para todas las comunicaciones.
* Implementación de **logs de actividad** para registrar accesos y procesos ETL.
* Validación y sanitización de datos cargados para evitar inyecciones o corrupción de datos.

### 3.3.3 Fiabilidad

* El sistema debe manejar errores en los procesos ETL y generar registros automáticos.
* Debe garantizar la consistencia de los datos cargados en la base de datos.

### 3.3.4 Disponibilidad

* El sistema debe mantener disponibilidad ≥ 95% mensual.
* En caso de caída del servidor, debe contar con procedimientos documentados de recuperación.

### 3.3.5 Mantenibilidad

* El sistema debe estar diseñado con código modular y documentado.
* Se deben generar reportes de ejecución de ETL semanalmente para revisión.

### 3.3.6 Portabilidad

* El backend debe estar desarrollado en Python/Django para facilitar despliegues en entornos Linux y Windows.
* El frontend debe ser compatible con navegadores modernos sin necesidad de plugins adicionales.

## 3.4 Otros Requisitos

* Posible incorporación futura de un módulo de forecasting para predicción de precios.
* Inclusión de exportación de datos a formatos Excel, CSV o PDF en versiones posteriores.
* Integración opcional con notificaciones por correo para alertar sobre cambios de precio.