

# **Documento de requerimientos de software**

***UniversalBoxApp***

***Fecha: 16 de abril de 25***

## Contenido

Historial de Versiones .....	3
Información del Proyecto.....	3
Aprobaciones .....	3
1. Propósito .....	4
2. Alcance del producto / Software .....	4
3. Referencias .....	5
4. Funcionalidades del producto.....	5
5. Clases y características de usuarios .....	6
6. Entorno operativo .....	6
7. Requerimientos funcionales .....	8
a. Funcionalidades para la carga y procesamiento de datos .....	8
b. Funcionalidad de la búsqueda semántica .....	9
c. Funcionabilidad de visualización - Dashboard .....	10
d. Otras funcionabilidades.....	11
8. Reglas de negocio.....	12
9. Requerimientos de interfaces externas .....	12
9.1. Interfaces de usuario.....	12
9.2. Interfaces de hardware .....	13
9.3. Interfaces de software.....	13
9.4. Interfaces de comunicación .....	13
10. Requerimientos no funcionales .....	13
11. Glosario .....	15

## Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
04 de abril de 2025	V1	David Aucancela	Epoch – Aplicaciones informáticas 2	Propósito del proyecto aplicativo
12 de abril de 2025	V1	David Aucancela	Epoch – Aplicaciones informáticas 2	Requisitos del proyecto
14 de abril de 2025	V1	David Aucancela	Epoch – Aplicaciones informáticas 2	Casos de uso

## Información del Proyecto

Empresa / Organización	Escuela superior politécnica de Chimborazo
Proyecto	Asistente de búsqueda semántica Universal Box
Fecha de preparación	04 de abril del 2025
Cliente	Universal Box
Patrocinador principal	Universal Box
Gerente / Líder de Proyecto	David Aucancela
Gerente / Líder de Análisis de negocio y requerimientos	David Aucancela

## Aprobaciones

Nombre y Apellido	Cargo	Departamento u Organización	Fecha	Firma
Julio Santillán	Docente	Facultad de informática y electrónica	14 de abril 2025	

## 1. Propósito

**Título:** Asistente de Búsqueda Semántica Universal Box

Transformar la forma en que se consulta y se analiza la información en la empresa Universal Box, superando las limitaciones de los métodos tradicionales basados en filtros de Excel.

Este sistema busca la implementación de inteligencia artificial, embeddings y procesamiento de lenguaje natural, con el objetivo innovar la manera en que se analiza la información en una empresa.

**Funcionabilidades:**

- Carga de archivos .xlsx a la base de datos.
- Agente inteligente para búsqueda semántica.
- Interacción de un Dashboard interactivo para realizar consultas y visualizar resultados.
- Gestión de usuarios (digitador, gerente y admin).

## 2. Alcance del producto / Software

**Objetivo principal**

Desarrollar un sistema inteligente que, mediante técnicas de embeddings y procesamiento de lenguaje natural, permita realizar búsquedas semánticas en la información empresarial, facilitando la toma de decisiones estratégicas y mejorando la competitividad de Universal Box.

**Beneficios**

- Pronto acceso a consultas complejas en lenguaje natural, reduciendo la obtención de datos para la toma de decisiones.
- Integridad y confiabilidad de la información cargada mediante la normalización de los archivos.
- Identificación de tendencias y patrones para estrategias corporativas,

### 3. Referencias

Desarrollo de un asistente conversacional basado en inteligencia artificial para atender trámites académicos de la carrera de software., Paucar M., Romero C. V1. 2024. Ecuador – Riobamba – ESPOCH.

Desarrollo de un chatbot basado en inteligencia artificial para atención al cliente de la farmacia San Vicente del cantón Chambo. Guaylla B., Burgos J. V1. 2024. Ecuador – Riobamba – ESPOCH.

Modelo de web semántica para universidades. Abad K., Ozrozco W., Torres W., Gonzales A. Revista Científica y Tecnológica UPSE, Vol. III, (Dic., 2015).

Cuadro declarativo 2024 – UniversalBox – Año 2024.

### 4. Funcionalidades del producto

#### a. Funcionalidades para la carga y procesamiento de datos

- Carga del archivo .xls validando el formato.
- Limpieza de los datos para la detección de errores y formato.
- Cargar y mapeo de la base de datos asegurando la persistencia de datos.

#### b. Funcionalidad de la búsqueda semántica

- Campo de búsqueda donde los usuarios puedan ingresar consultas en lenguaje natural.
- Crear vectorizaciones de texto mediante la generación de embeddings para cada registro o documento almacenado.
- Indexación de los embeddings en el motor de búsqueda en una base de datos vectorial.
- Capacidad para integrar nuevos datos y actualizar los embeddings conforme se obtenga nueva información.

#### d. Funcionabilidad de visualización - Dashboard

- Representar los datos en tablas mediante un dashboard interactivo, de forma que el usuario pueda analizarlos rápidamente.

- Visualización de resultados ordenados mediante el usuario necesite.

#### c. Otras funcionalidades

- Gestión de roles y permisos para administrar quién puede subir datos, acceder a la información y realizar las consultas.
- Registro de las operaciones realizadas (subidas de archivos, búsquedas) para auditoría.

## 5. Clases y características de usuarios

Usuario	Frecuencia de uso	Funcionabilidades	Privilegios	Nivel de experiencia
Admin	Muy frecuente	Todas	Todos	Alto
Digitador	Frecuente	Carga de archivos	Carga de archivos	Bajo
Gerente	Poco frecuente	Carga de archivos, consultas y visualización	Realizar consultas	Bajo

## 6. Entorno operativo

El entorno operativo en el que se desplegará la aplicación debe ser flexible a los cambios y mejoras que se vayan realizando en la creación del sistema, capaz de ofrecer escalabilidad según la demanda.

#### Plataforma hardware:

- **Dispositivo físico:** Hp i7-7ma 2.7GHz
- **Ram:** 24 GB.
- **Almacenamiento:** SSD 550gb

### **Sistema operativo:**

- **Sistema operativo:** Windows 10 home
- **Procesador:** 64 bits, x64

### **Plataformas software:**

- **Frontend:** Angular
- **Backend:** Python/Django
- **Carga de .xls:** Python Pandas
- **Procesamiento de lenguaje natural:** spaCy
- **Base de datos multivectorial:** Qdrant o Alternativa
- **Servicios de autenticación:** Integración con django rest framework (DRF)
- **Herramientas de monitoreo y logging:** JWT (JSON Web Tokens)

### **Red y Seguridad:**

- Conectividad de alta velocidad en la red
- Políticas de seguridad y firewalls de Windows defender

## 7. Requerimientos funcionales

### a. Funcionalidades para la carga y procesamiento de datos

Identificación	Sub identificación	Nombre del requisito	Descripción del requisito	Versión	Estado actual	Criterios de aceptación	Nivel de complejidad	Nivel de prioridad	Comportamiento del sistema	Condiciones de error
RF	1	<b>Carga de archivo</b>	El sistema permitirá cargar un archivo de formato .xlsx o formatos semejantes, en el sistema	v1	Por desarrollar	El tipo de formato del archivo es correcto, el archivo pesa el máximo permitido, carga del archivo sin problemas de red	Media	Alta	Mensaje de carga exitosa	Mensaje de error identificado al cargar el archivo
RF	2	<b>Identificación de columnas</b>	El sistema realizará la identificación de todas las columnas disponibles del archivo cargado	v1	Por desarrollar	El archivo contiene las columnas necesarias, verificar duplicidad en las columnas	Baja	Alta	Mensaje de columnas identificadas	Mensaje de tipo de columnas no disponibles
RF	3	<b>Limpieza de datos</b>	El sistema realizará la limpieza de la información utilizando procesos ETL.	v1	Por desarrollar	Los datos se encuentran completos, los datos se encuentran disponibles, se establece un formato único de dato	Alta	Media	Mensaje de datos listos para usar	Mensaje de información no disponible



## Escuela superior politécnica de Chimborazo

RF	4	<b>Población de la base de datos</b>	El sistema realizará la población de la base de datos con la información del archivo cargado, este proceso se asigna mediante un ORM propio de Django	v1	Por desarrollar	Las consultas realizadas en la base de datos se verifican con la información de los archivos.	Alta	Alta	Persistencia de la información	No persistencia de la información
----	---	--------------------------------------	---	----	-----------------	---	------	------	--------------------------------	-----------------------------------

### b. Funcionalidad de la búsqueda semántica

Identificación	Sub identificación	Nombre del requisito	Descripción del requisito	Versión	Estado actual	Criterios de aceptación	Nivel de complejidad	Nivel de prioridad	Comportamiento del sistema	Condiciones de error
RF	5	<b>Campo de búsqueda</b>	El sistema permitirá que se puedan realizar búsquedas avanzadas mediante el lenguaje natural - NLP	v1	Por desarrollar	Ingreso de cadena de caracteres con controles, opción para eliminar el texto ingresado, búsquedas recientes	Alta	Alta	Visualización de la información solicitada	Mensaje de error en la solicitud
RF	6	<b>Generación de los embeddings</b>	El sistema permitirá la vectorización de la consulta semántica	v1	Por desarrollar	Se asigna los valores al vector para realizar la transformación de los embeddings, la consulta debe tener un control de tamaño para el control de los tokens	Alta	Alta	Verificación de la cantidad de tokens de la solicitud	Mensaje de consulta no realizada

## Escuela superior politécnica de Chimborazo

RF	7	<b>Indexación en el motor de búsqueda</b>	Entrenamiento del modelo a utilizar para el razonamiento avanzado	v1	Por desarrollar	Entrenamiento del modelo de procesamiento de lenguaje natural para la realización de las consultas	Alta	Alta	Predicción mediante la consulta realizada	Mensaje de entrenamiento no permitido
RF	8	<b>Actualización de los embeddings</b>	El sistema actualizará los embeddings cada que se realice la carga de un nuevo archivo	v1	Por desarrollar	Se asigna los valores float al vector para realizar la transformación de los embeddings, la consulta debe tener un control de tamaño para el control de los tokens	Alta	Media	Predicción mediante la consulta realizada	Mensaje de entrenamiento no permitido

### c. Funcionabilidad de visualización - Dashboard

Identificación	Sub identificación	Nombre del requisito	Descripción del requisito	Versión	Estado actual	Criterios de aceptación	Nivel de complejidad	Nivel de prioridad	Comportamiento del sistema	Condiciones de error
RF	9	<b>Visualización de la consulta</b>	El sistema mostrará la información solicitada en un dashboard de forma rápida y accesible al usuario	v1	Por desarrollar	Se muestra la información mediante elementos interactivos, siguiendo principios UX	Alta	Alta	Verificación del dashboard con la información solicitada	Mensaje de error de la información, información no disponible

RF	10	Ordenamiento de la información	El sistema permitirá categorizar la información solicitada mediante la preferencia del usuario	v1	Por desarrollar	La información se puede categorizar mediante un modelo de ordenamiento ingresado por el usuario	Media	Media	Verificación de la información ordenada	Mensaje de error, método de ordenamiento no válido
----	----	--------------------------------	--	----	-----------------	---	-------	-------	---	--

#### d. Otras funcionalidades

Identificación	Sub identificación	Nombre del requisito	Descripción del requisito	Versión	Estado actual	Criterios de aceptación	Nivel de complejidad	Nivel de prioridad	Comportamiento del sistema	Condiciones de error
RF	11	Gestión de usuarios	El sistema permitirá realizar el registro de los distintos usuarios del sistema, cada usuario tiene sus propias responsabilidades	v1	Por desarrollar	Se puede escoger entre las opciones de usuario disponibles (admin, gerente, digitador)	Medio	Alta	Creación del usuario	Mensaje de error, el usuario no se encuentra disponible
RF	12	Registro de operaciones	El sistema permitirá registrar los archivos cargados y las consultas realizadas	v1	Por desarrollar	La información previamente solicitada llevará un registro por fecha	Alta	Alta	Verificación de la información registrada	Mensaje de error, información no registrada

## **8. Reglas de negocio**

Todos los usuarios deben completar una autenticación multifactorial antes de acceder al sistema.

Los usuarios serán asignados a segmentos de red específicos según su rol, restringiendo el acceso a datos sensibles según el nivel de autorización.

Si un usuario está inactivo durante más de 15 minutos, su sesión debe expirar automáticamente, obligando al usuario a volver a autenticarse para continuar con la operación.

Todos los datos sensibles, como contraseñas, deben ser cifrados tanto en tránsito como en reposo para garantizar la protección contra accesos no autorizados.

## **9. Requerimientos de interfaces externas**

### **9.1. Interfaces de usuario**

- El sistema tendrá diferentes interfaces según el rol de usuario (administrador, gerente, digitador) y las funcionalidades disponibles para cada uno.
- Ejemplos visuales de las pantallas principales, como:
  - Pantalla de inicio de sesión con autenticación multifactorial.
  - Pantalla principal de transacciones con filtros y búsqueda de operaciones.
  - Dashboard de administración con vistas de monitoreo de actividades.
- El diseño de la interfaz debe ser intuitivo y fácil de usar.
- Uso consistente de colores, tipografías y botones que sigan un estilo corporativo o de producto.
- Asegurar que la interfaz sea responsiva para diferentes tamaños de pantalla (computadora, tablet, móvil).
- Disposición clara de los elementos en la pantalla para mejorar la usabilidad.

### 9.2. Interfaces de hardware

- El sistema debe ser accesible desde navegadores web comunes (Chrome, Firefox, Edge) en computadoras de escritorio y laptops.
- Protocolos Ethernet/Wi-Fi permitidos para la conexión con servidores y otros dispositivos de red.
- El sistema debe ser capaz de interactuar con dispositivos de hardware para la transmisión de datos (teclado, mouse).

### 9.3. Interfaces de software

- El sistema interactúa con bases de datos SQL - PostgreSQL para almacenar y recuperar información de transacciones y usuarios.
- El sistema debe ser compatible con sistemas operativos modernos como Windows, Linux y macOS.
- El sistema utilizará frameworks como Django para el backend, React para el frontend, y Celery para manejar tareas en segundo plano.

### 9.4. Interfaces de comunicación

- El sistema debe ser capaz de enviar correos electrónicos para restablecer la contraseña.
- El sistema debe ser accesible a través de navegadores web con soporte para HTTP/HTTPS.
- XML para la estructura de los datos transmitidos en las API del sistema.
- Autenticación para integrar servicios externos de manera segura y garantizar que solo usuarios autorizados accedan a las APIs.

## 10. Requerimientos no funcionales

### Requerimientos No Funcionales Generales:

1. El sistema debe ser capaz de manejar hasta 100 usuarios concurrentes sin degradar la experiencia del usuario. Las respuestas a las peticiones del sistema deben ser procesadas en menos de 10 segundos.

2. El sistema debe ser escalable horizontalmente para adaptarse al crecimiento de usuarios y volumen de datos.
3. El sistema debe garantizar una disponibilidad del 99.9%, con un tiempo de inactividad planificado de máximo 4 horas al mes para mantenimiento.
4. El sistema debe ser accesible desde los navegadores web más utilizados (Chrome, Firefox, Edge) y en dispositivos móviles a través de un navegador compatible con HTML5.
5. La interfaz de usuario debe ser intuitiva y fácil de navegar, con un diseño limpio y coherente. El sistema debe estar diseñado siguiendo las mejores prácticas de UX/UI para asegurar una experiencia fluida.
6. El código del sistema debe seguir estándares de codificación bien definidos, con documentación adecuada y comentarios para facilitar el mantenimiento y futuras actualizaciones.
7. El sistema debe ser capaz de integrarse con servicios externos a través de APIs RESTful de forma eficiente, utilizando formatos de intercambio de datos como JSON o XML.
8. El sistema debe cumplir con las normativas legales de protección de datos, como GDPR para la gestión de datos personales, asegurando que todos los datos almacenados y procesados sean tratados de manera segura y conforme a las leyes aplicables.
9. El frontend debe ser desarrollado con Angular 12 o una versión superior, utilizando una arquitectura modular que permita el fácil mantenimiento y expansión de las funcionalidades.
10. El backend debe estar construido utilizando Django 3.2 o superior, implementando patrones de diseño como MVC (Modelo-Vista-Controlador), y debe ser capaz de manejar peticiones RESTful a través de Django REST Framework (DRF).
11. El sistema debe ser capaz de cargar archivos .xls de hasta 100 MB utilizando Python Pandas, con validación de datos para asegurar la integridad de los datos importados y manejo eficiente de grandes volúmenes de información.
12. El sistema debe integrar spaCy para el procesamiento de lenguaje natural, permitiendo la extracción de entidades, análisis de sentimientos, y

clasificación de texto. El tiempo de procesamiento de cada texto no debe exceder los 5 segundos por documento.

13. El sistema debe usar una base de datos multivectorial como Qdrant o una alternativa que permita el almacenamiento y búsqueda eficiente de vectores. Las consultas deben devolverse en menos de 1 segundo para conjuntos de datos de hasta 1 millón de registros.
14. El sistema debe implementar JWT (JSON Web Tokens) para la autenticación y autorización de usuarios. Los tokens deben caducar automáticamente después de 1 hora de inactividad y ser renovados mediante un proceso seguro.
15. El sistema debe integrar herramientas de monitoreo como Prometheus y Grafana para el análisis de rendimiento y la supervisión en tiempo real. Además, debe utilizar JWT para la autenticación de las API, garantizando que los tokens sean seguros y se mantengan encriptados.

## 11. Glosario

**Autenticación Multifactorial (MFA):** Método de seguridad que requiere dos o más factores de verificación para acceder a un sistema, como una contraseña y un código enviado por SMS.

**JWT (JSON Web Token):** Estándar abierto para la transmisión segura de información entre dos partes como un objeto JSON. Es utilizado principalmente para la autenticación de usuarios en sistemas web.

**SPA (Single Page Application):** Aplicación web que carga una sola página HTML y actualiza dinámicamente el contenido sin recargar la página completa.

**Django REST Framework (DRF):** Framework para construir APIs RESTful en Django, proporcionando herramientas y componentes para crear aplicaciones web y móviles eficientes.

**Base de Datos Multivectorial:** Base de datos diseñada para almacenar y gestionar datos en forma de vectores, optimizada para la búsqueda de similitudes y análisis de datos complejos como los de aprendizaje automático.

**spaCy:** Biblioteca de procesamiento de lenguaje natural (NLP) de código abierto, utilizada para realizar tareas como el análisis gramatical, extracción de entidades y modelado de texto.

**Pandas:** Biblioteca de Python para la manipulación y análisis de datos, especialmente útil para trabajar con archivos de tipo **.xls** o **.csv** y grandes volúmenes de información tabular.

**Qdrant:** Base de datos multivectorial que permite almacenar y realizar búsquedas eficientes de vectores de alta dimensión, usada comúnmente en sistemas de recomendación y búsqueda semántica.

**Token de Acceso:** Clave generada que permite al usuario o sistema acceder a recursos específicos dentro de una API o sistema web durante un tiempo determinado.

**Entidades (en NLP):** Elementos clave extraídos del texto en el procesamiento de lenguaje natural, como nombres de personas, lugares, fechas, etc.

**Búsqueda Semántica:** Técnica de búsqueda que utiliza el significado y contexto de las palabras en lugar de la coincidencia exacta de los términos. Permite resultados más relevantes basados en el contexto del texto.

**Embeddings:** Representaciones vectoriales de palabras, frases o documentos en un espacio de alta dimensión. Se utilizan en procesamiento de lenguaje natural (NLP) para capturar el significado semántico de las palabras.

**Modelos de Transformer:** Arquitectura de redes neuronales utilizada en modelos de procesamiento de lenguaje como **BERT** y **GPT**. Utiliza mecanismos de atención para comprender el contexto y las relaciones entre palabras en un texto.