|  |  |  |
| --- | --- | --- |
| Una caricatura de una persona  Descripción generada automáticamente con confianza baja | UNIVERSIDAD DE BURGOS  ESCUELA POLITÉCNICA SUPERIOR  Grado en Ingeniería Informática |  |

**TFG del Grado en Ingeniería Informática**

**<** **Sistema de extracción y gestión automatizada de requisitos desde documentos PDF para proyectos de ingeniería >**

**Documentación Técnica**

Presentado por Estíbalitz Díez

en Universidad de Burgos

15 de enero de 2025

**Tutores:** Dr. José Manuel Galán Ordax y Dr. José Ignacio Santos Martin

Índice general

[Índice general III](#_Toc182683104)

[Índice de figuras V](#_Toc182683105)

[Índices de tablas VI](#_Toc182683106)

[*Apéndice A-* Plan de proyecto Software 1](#_Toc182683107)

[A.1. Introducción 1](#_Toc182683108)

[A.2. Planificación temporal 1](#_Toc182683109)

[A.2.1 Sprints 1](#_Toc182683110)

[A.3. Estudio de viabilidad 6](#_Toc182683111)

[*Apéndice B-* Especificación de Requisitos 8](#_Toc182683112)

[B.1. Introducción 9](#_Toc182683113)

[B.2. Objetivos generales 9](#_Toc182683114)

[B.3. Catálogo de requisitos 10](#_Toc182683115)

[B.3.1 Requisitos Funcionales 10](#_Toc182683116)

[B.3.2 Requisitos No funcionales 11](#_Toc182683117)

[B.4. Especificación de requisitos 13](#_Toc182683118)

[B.4.1 Casos de uso 13](#_Toc182683119)

[*Apéndice C-* Especificación de diseño 19](#_Toc182683120)

[C.1. Introducción al diseño 20](#_Toc182683121)

[C.2. Diseño de datos 20](#_Toc182683122)

[C.2.1 Estructura de datos 20](#_Toc182683123)

[C.2.2 Diagrama E/R 22](#_Toc182683124)

[C.2.3 Diagrama Relacional 23](#_Toc182683125)

[C.3. Diseño procedimental 24](#_Toc182683126)

[C.3.1 Diagrama secuencias Cargar 24](#_Toc182683127)

[C.3.2 Diagrama secuencias Consulta 25](#_Toc182683128)

[C.3.3 Diagrama secuencia Acciones 26](#_Toc182683129)

[C.4. Diseño arquitectónico 27](#_Toc182683130)

[C.4.1 Descripción General de la Arquitectura 27](#_Toc182683131)

[C.4.2 Componentes Principales 27](#_Toc182683132)

[C.4.3 Flujo de Datos 27](#_Toc182683133)

[C.4.4 Patrones de Diseño Utilizados 27](#_Toc182683134)

[C.4.5 Patrón Repository (simplificado): 27](#_Toc182683135)

[*Apéndice D-* Documentación técnica de programación 28](#_Toc182683136)

[D.1. Introducción 29](#_Toc182683137)

[D.2. Estructura de directorios 29](#_Toc182683138)

[D.3. Manual del programador 29](#_Toc182683139)

[D.4. Pruebas del sistema 29](#_Toc182683140)

[*Apéndice E-* Documentación de usuario 30](#_Toc182683141)

[E.1. Introducción 31](#_Toc182683142)

[E.2. Requisitos de usuario 31](#_Toc182683143)

[E.3. Instalación 31](#_Toc182683144)

[E.4. Manual del usuario 31](#_Toc182683145)

[*Apéndice F-* Anexo de sostenibilización curricular 32](#_Toc182683146)

[F.1. Introducción 33](#_Toc182683147)

[F.2. Competencias en sostenibilidad adquiridas 33](#_Toc182683148)

[F.3. Impacto social y ambiental 33](#_Toc182683149)

[Bibliografía 35](#_Toc182683150)

Índice de figuras

[Figura 1 Sprint 1 2](#_Toc181814732)

[Figura 2 Sprint 2 2](#_Toc181814733)

[Figura 3 Sprint 3 3](#_Toc181814734)

[Figura 4 Sprint 4 3](#_Toc181814735)

[Figura 5 Sprint 5 4](#_Toc181814736)

[Figura 6 Sprint 6 4](#_Toc181814737)

[Figura 7 Sprint 7 5](#_Toc181814738)

[Figura 8 Diagrama de casos de uso 13](#_Toc181814739)

[Figura 9 Diagrama Entidad - Relación 22](#_Toc181814740)

[Figura 10 Diagrama relacional 23](#_Toc181814741)

[Figura 11 Diagrama Secuencia CARGAR 24](#_Toc181814742)

[Figura 12 Diagrama Secuencia CONSULTA 25](#_Toc181814743)

[Figura 13 Diagrama Secuencia Otras acciones 26](#_Toc181814744)

[Figura F.3.1 Imagen ejemplo 34](#_Toc181814745)

Índices de tablas

[Tabla 1 Requisitos Funcionales 10](#_Toc181814720)

[Tabla 2 Cargar Requisitos – CU 1- Importación de requisitos 14](#_Toc181814721)

[Tabla 3 Cargar Requisitos – CU 2 – Asociación de proyectos 14](#_Toc181814722)

[Tabla 4 Cargar Requisitos – CU 3 – Extracción de capítulos y requisitos 15](#_Toc181814723)

[Tabla 5 Cargar Requisito – CU 4 – Edición de requisitos 15](#_Toc181814724)

[Tabla 6 Cargar Requisitos – CU 5 – Almacenamiento en BBDD 16](#_Toc181814725)

[Tabla 7 Cargar Requisitos – CU 6 – Validación de Asociaciones 16](#_Toc181814726)

[Tabla 8 Consulta CU 7 – Consulta de documentos y Requisitos 17](#_Toc181814727)

[Tabla 9 Consulta CU 8- Exportación de requisitos 17](#_Toc181814728)

[Tabla 10 Actualización BBDD CU 9 - Actualización 18](#_Toc181814729)

[Tabla 11 Idioma CU 10 – Manejo Multilingüe 18](#_Toc181814730)

[Tabla 12 Tabla modelo 34](#_Toc181814731)

# Plan de proyecto Software

## Introducción

El objetivo de este proyecto es desarrollar una aplicación de escritorio para la gestión de requisitos de ingeniería que permita el procesamiento automatizado de documentos PDF, la identificación de información relevante y la organización de requisitos asociados a proyectos específicos. Esta herramienta está diseñada para ser una plataforma de gestión accesible y escalable, que facilite la extracción de datos y su asociación a subsistemas mediante palabras clave, proporcionando a los usuarios una forma intuitiva de gestionar y consultar documentos de requisitos.

Además de servir como herramienta de gestión, la aplicación permite la exportación de datos en formato CSV, haciéndola compatible con otros sistemas avanzados de gestión de requisitos, como IBM DOORS. Esto asegura que la información almacenada en el sistema pueda ser utilizada en otros entornos, aumentando su aplicabilidad y valor como plataforma intermedia.

Debido a la naturaleza del proyecto, es decir, que corresponde al Trabajo de Fin de Grado de una alumna del grado, en este caso de la autora, el desarrollo del proyecto se ha realizado por un único componente y en el plazo de tiempo que establece la normativa de la UBU una vez aprobada la asignación del tema propuesto.

## Planificación temporal

La Planificación Temporal de este proyecto se ha diseñada para organizar y distribuir las tareas de desarrollo en un calendario claro y estructurado, siguiendo los requisitos de metodología SCRUM establecidos por la UBU.

Para lograrlo se ha generado el proyecto en el software Zube vinculado a un repositorio creado con el propósito de almacenar el código en el repositorio GitHub y poder registrar los avances mediante commits en Git garantizando así la homogeneidad en las actualizaciones y avances.

La planificación ha sido organizada en sprints de una duración aproximada de 1 semana cada uno, con un propósito general y con tareas específicas en cada uno de ellos a desarrollar.

Además, se establecieron unos hitos como referencias temporales con ciertas holguras para poder garantizar el cumplimiento de entrega en la fecha solicitada por la UBU y disponer con tiempo extra para poder realizar posteriores versiones o mejoras del software.

El trabajo de análisis, búsqueda de herramientas y bibliotecas se inició con anterioridad cuando la idea se valoraba como posible propuesta.

El 20 de septiembre fue la primera tutoría y por tanto, es la fecha que contamos como inicio del TFG, aunque oficialmente la aprobación del tema fue unos días más tarde. Mis tutores me aconsejaron el uso de Zube y Github, así como la lectura de documento no oficial con recomendaciones, de respuestas a preguntas frecuentes de la plataforma de la UBU y del TFG de Rodrigo Merino Tovar.

### Sprints

Los Sprints comienzan el 21 de septiembre con tareas específicas de arranque y preparación con los primeros pasos.del programa hasta el 3 de noviembre que es cuando comienzan las tareas de generación del contenido de documentación. A continuación se detallan cada uno de los sprints:

#### Sprint 1 – Estudio y análisis de herramientas / Tokenes

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Figura 1 Sprint 1

* **Duración**: 21 – 28 de septiembre
* **Objetivo**: Definir las herramientas y confeccionar las plantillas
* **Contexto**: Tras la reunión con los tutores, se identifican ciertas herramientas necesarias para el desarrollo de TFG.
* **Tareas**:
* Lectura y revisión de la documentación aconsejada por los tutores.
* Creación de cuentas, instalación y familiarización de las herramientas.
* Creación de las plantillas.
* Generación de versión 0 de la documentación.
* Celebración de la reunión para definición de tokenes para la detección del alcance de los requisitos.

#### Sprint 2 – BBDD – Extracción

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Figura 2 Sprint 2

* **Duración**: 23 – 28 de septiembre
* **Objetivo**: xxx
* **Contexto**: xxx
* **Tareas**: xxx

#### Sprint 3 - Interfaz

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Figura 3 Sprint 3

* **Duración**: 28 de septiembre – 4 de octubre
* **Objetivo**: xxx
* **Contexto**: xxx
* **Tareas**: xxx

#### Sprint 4 - Asignación

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Figura 4 Sprint 4

* **Duración**: 5 – 10 de octubre
* **Objetivo**: xxx
* **Contexto**: xxx
* **Tareas**: xxx

#### Sprint 5 – Consulta y Load V2

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Figura 5 Sprint 5

* **Duración**: 11 – 17 de octubre
* **Objetivo**: xxx
* **Contexto**: xxx
* **Tareas**: xxx

#### Sprint 6 – Idiomas / Ajustes / Correcciones / Test

Gráfico

Descripción generada automáticamente

Figura 6 Sprint 6

* **Duración**: 18 de octubre – 3 de noviembre
* **Objetivo**: xxx
* **Contexto**: xxx
* **Tareas**: xxx

#### Sprint 7 Documentación I

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Figura 7 Sprint 7

* **Duración**: 3 – 10 de noviembre
* **Objetivo**: xxx
* **Contexto**: xxx
* **Tareas**: xxx

#### Sprint 8 – Documentación II

* **Duración**: 10 – 17 de noviembre
* **Objetivo**: xxx
* **Contexto**: xxx
* **Tareas**: xxx

#### Sprint 9 Mejora V1 / V2

* **Duración**: 18 – 24 de noviembre
* **Objetivo**: xxx
* **Contexto**: xxx
* **Tareas**: xxx

## Estudio de viabilidad

El desarrollo de la aplicación de gestión de requisitos de ingeniería ha sido concebido para ofrecer una solución sencilla, flexible y escalable que permita a los usuarios cargar, analizar y gestionar documentos de requisitos en formato PDF. A continuación, se detallan los aspectos técnicos, económicos, operativos y de recursos que justifican la viabilidad de este proyecto y aseguran su efectividad para cubrir las necesidades identificadas.

Viabilidad Técnica

Desde el punto de vista técnico, la aplicación utiliza tecnologías accesibles y bien soportadas, como Python, SQLite y Tkinter, que garantizan tanto la estabilidad del sistema como la facilidad de mantenimiento. La elección de SQLite como base de datos permite un almacenamiento local y eficiente de los documentos, requisitos y asociaciones de subsistemas, con la flexibilidad de poder migrar a una base de datos más robusta si el volumen de datos lo requiere en el futuro.

La aplicación está diseñada para ser compatible con sistemas operativos Windows y macOS, lo que la hace accesible en distintas plataformas sin necesidad de adaptaciones complejas. Además, la modularidad de su estructura, con archivos de configuración independientes para las palabras clave y los idiomas, asegura una fácil personalización y expansión futura, tanto para el análisis de textos como para el soporte multilingüe y la ayuda integrada en distintos idiomas.

Viabilidad Económica

El uso de tecnologías de código abierto, como Python y SQLite, reduce significativamente los costos de desarrollo, ya que no es necesario adquirir licencias costosas para implementar la aplicación. La mayor inversión radica en las horas de desarrollo y en la fase de pruebas, que requieren una dedicación cuidadosa para garantizar que el sistema funcione correctamente y cumpla con los requisitos planteados. La posibilidad de exportar los datos a formatos CSV compatibles con otras herramientas de gestión de requisitos, como IBM DOORS, añade valor al sistema, aumentando su aplicabilidad y justificación económica al ser útil como plataforma intermedia.

Viabilidad Operativa

La aplicación se ha diseñado teniendo en cuenta las necesidades de los usuarios en la gestión de requisitos de ingeniería. La interfaz, desarrollada con Tkinter, es sencilla e intuitiva, permitiendo que los usuarios puedan cargar documentos, validar asociaciones y realizar consultas de forma rápida y sin necesidad de capacitación extensiva. La capacidad de exportación a CSV asegura que los datos pueden ser utilizados en otros sistemas de gestión de requisitos, ampliando la utilidad de la herramienta en diferentes entornos operativos. Asimismo, el sistema de palabras clave y tokens es fácilmente ampliable, permitiendo ajustar la aplicación a nuevas áreas de análisis sin reprogramaciones complejas.

Viabilidad de Recursos

Para el desarrollo de esta aplicación, los recursos necesarios se limitan a un entorno de desarrollo Python, soporte para pruebas en plataformas Windows y macOS, y acceso a herramientas de gestión de bases de datos. La disponibilidad de desarrolladores con conocimientos en Python y manipulación de datos en SQLite permite abordar el proyecto sin requerir perfiles técnicos altamente especializados. Además, el uso de archivos de configuración independientes para los tokens, idiomas y ayudas reduce la carga de mantenimiento y permite realizar actualizaciones sin necesidad de modificar el código principal.

# Especificación de Requisitos

## Introducción

La Especificación de Requisitos de esta aplicación de gestión de requisitos de ingeniería tiene como objetivo definir de manera precisa y detallada las funcionalidades y características esenciales que el sistema debe cumplir. Este capítulo describe los requisitos necesarios para asegurar que la aplicación no solo cumpla con las funciones básicas esperadas, sino también con ciertos estándares de rendimiento, escalabilidad, y facilidad de uso.

El sistema está concebido como una herramienta inicial de gestión de documentos y requisitos, pensada para integrarse como una plataforma intermedia que permita el procesamiento de información y la exportación a otros sistemas avanzados de gestión de requisitos, como IBM DOORS. La especificación de requisitos se ha estructurado en dos categorías principales: requisitos funcionales, que describen las funcionalidades específicas que debe proporcionar el sistema, y requisitos no funcionales, que definen las características de calidad que aseguran un rendimiento eficiente, una interfaz intuitiva y una infraestructura preparada para el crecimiento y la interoperabilidad.

La especificación de requisitos detallada en este documento permitirá guiar el desarrollo y la evaluación de la aplicación, asegurando que cumpla con las necesidades del usuario final y facilite su expansión y compatibilidad con otras plataformas en el futuro.

## Objetivos generales

El objetivo principal de este proyecto es desarrollar una aplicación de escritorio para la gestión de requisitos de ingeniería, que facilite la carga y el procesamiento de documentos PDF para extraer y organizar información clave en capítulos y requisitos. La herramienta está diseñada para simplificar el análisis de textos y la asociación automática con subsistemas mediante el uso de palabras clave, permitiendo a los usuarios revisar y validar estas asociaciones antes de almacenarlas.

Entre los objetivos generales del proyecto se incluyen:

* Proveer una plataforma de gestión de requisitos que permita el registro, organización y consulta de documentos asociados a proyectos específicos.
* Facilitar la extracción automática de información de documentos PDF y su organización en capítulos y requisitos, optimizando el tiempo de procesamiento y mejorando la eficiencia del análisis documental.
* Ofrecer un sistema de análisis de palabras clave y asociaciones con subsistemas, permitiendo la identificación automática de los alcances de cada documento / requisitos y simplificando el proceso de validación.
* Permitir la exportación de datos en formatos compatibles (CSV), asegurando que la información almacenada sea interoperable con otras herramientas de gestión de requisitos.
* Desarrollar una arquitectura modular y escalable, diseñada para facilitar el crecimiento de la base de datos y la adición de nuevas funcionalidades, así como el soporte multilingüe y la integración de ayudas en distintos idiomas.

Esta aplicación representa una solución inicial en la gestión de requisitos de ingeniería, con un diseño pensado para futuras expansiones y compatibilidad con sistemas avanzados. A través de estos objetivos, se busca proporcionar una herramienta eficiente, flexible y adaptable a distintas necesidades de gestión documental en el ámbito de la ingeniería.

## Catálogo de requisitos

### Requisitos Funcionales

| COD | TITULO | DESCRIPCION | | | |
| --- | --- | --- | --- | --- | --- |
| RF1 | **Carga de documentos PDF** | **El sistema debe permitir al usuario cargar un documento PDF y asociarlo a un proyecto específico.** | | | |
|  |  | RF1.1 | Asociación de Documentos a Proyectos | | El sistema debe permitir seleccionar un proyecto existente o crear uno nuevo al momento de asociar un documento PDF. |
|  |  | RF1.2 | Extracción de Capítulos y Requisitos | | El sistema debe extraer automáticamente la estructura de capítulos y requisitos desde el documento PDF. |
|  |  | RF1.3 | Edición de los requisitos | | El sistema debe permitir editar los requisitos extraídos antes de su almacenado para validar el contenido por el usuario. |
|  |  | RF1.4 | Almacenamiento de Datos en la Base de Datos | | El sistema debe almacenar el documento, los requisitos y sus asociaciones validadas en una base de datos local. |
|  |  | RF1.5 | Actualización de Documentos y Requisitos | | El sistema debe permitir añadir información de documentos y requisitos, actualizando la base de datos. |
|  |  | RF1.6 | Eliminación de Documentos y Asociaciones | | El sistema debe permitir la eliminación de documentos, junto con sus requisitos y asociaciones con subsistemas. |
| RF2 | **Consulta de documentos / requisitos** | **El sistema debe permitir al usuario realizar consultas de documentos y requisitos filtrando por proyecto, subsistema o fecha de creación.** | | | |
|  |  | RF2.1 | Consultar requisitos, documentos, proyectos y subsistemas. | | El sistema debe permitir localizar un elemento en concreto o bien un listado completo. |
|  |  | RF2.2 | Filtrado por documentos, proyectos y subsistemas. | | El sistema debe facilitar la localización de los requisitos y documentos para un proyecto en concreto y para un subsistema en concreto. |
| RF3 | **Exportación resultados** | **El sistema debe permitir la exportación de los resultados de las consultas.** | | | |
|  |  | RF3.1 | Exportación de requisitos, documento, proyectos o subsistemas. | | El sistema debe permitir exportar los datos seleccionados en un archivo fácilmente compatible con otros softwares para su posterior importación o manipulación. . |
| RF4 | **Detección de alcance** | **El sistema debe analizar cada capítulo y requisito buscando palabras clave que identifiquen los subsistemas relevantes.** | | | |
|  |  | RF4.1 | Validación de Asociaciones con Subsistemas | | El sistema debe permitir al usuario validar y ajustar las asociaciones de subsistemas para cada requisito antes de guardar los datos. |
|  |  | RF4.2 | Actualización de palabras clave | | El sistema debe permitir la actualización y modificación de las palabras claves de asociación con los diferentes subsistemas. |
| RF 5 | **Manejo multilingüe** | **El sistema debe permitir al usuario trabajar en distintos idiomas, posibilitando el manejo y gestión de proyectos internacionales.** | | | |
|  |  | RF5.1 | Uso del software en distintos idiomas | El sistema debe estar preparado para usuarios de distintas nacionalidades. | |

Tabla 1 Requisitos Funcionales

### Requisitos No funcionales

* **RNF1. Compatibilidad como Plataforma de Gestión e Interoperabilidad**
* Descripción: La aplicación debe ser compatible tanto como una plataforma de gestión de requisitos independiente como una plataforma intermedia que permite la exportación de datos en formato CSV. Estos datos deben ser fácilmente importables a otros sistemas de gestión de requisitos, como IBM DOORS.
* Criterio de Aceptación: Los datos pueden exportarse en formato CSV sin pérdida de información, y el formato es compatible con la importación en sistemas de gestión como DOORS.
* **RNF2. Escalabilidad y Flexibilidad de la Base de Datos**
* Descripción: La aplicación utiliza SQLite como base de datos, pero está diseñada para permitir la fácil migración a bases de datos más robustas conforme crezcan los requerimientos de almacenamiento. La estructura de datos debe facilitar la conversión sin una reestructuración significativa.
* Criterio de Aceptación: La base de datos puede migrarse a otros sistemas SQL sin necesidad de rediseñar el esquema de datos.
* **RNF3. Modularidad y Facilidad de Expansión de Tokens y Palabras Clave**
* Descripción: Los tokens y palabras clave utilizados para identificar subsistemas y requisitos específicos deben estar almacenados en un sistema que permita la modificación fácilmente. Esto permite la ampliación del conjunto de palabras clave sin necesidad de cambiar el código fuente.
* Criterio de Aceptación: Los tokens pueden añadirse, eliminarse o modificarse directamente en los archivos de configuración sin afectar la funcionalidad de la aplicación.
* **RNF4. Soporte Multilingüe y Expansión de Idiomas**
* Descripción: La aplicación debe ofrecer soporte en varios idiomas y permitir la fácil adición de más idiomas. Las información relativa a cada idioma deben estar almacenada de forma independiente para facilitar su mantenimiento y expansión.
* Criterio de Aceptación: Los archivos de idioma son independientes y modificables, permitiendo agregar idiomas adicionales sin modificar el código principal.
* **RNF5. Ayuda Integrada y Multilingüe**
* Descripción: La aplicación debe incluir un sistema de ayuda integrada en los idiomas disponibles, con la capacidad de añadir contenido en otros idiomas de manera sencilla. La documentación de ayuda debe estar en archivos separados para facilitar la actualización y expansión.
* Criterio de Aceptación: La ayuda integrada es accesible en los idiomas soportados, y los archivos de ayuda pueden actualizarse o ampliarse independientemente del código.
* **RNF6. Diseño para Escalabilidad y Crecimiento**
* Descripción: La aplicación está concebida como una versión inicial, con una arquitectura y estructura de datos que permiten una expansión sencilla. Debe poder ampliarse para incluir más funcionalidades y soportar un mayor volumen de datos sin requerir una reestructuración significativa.
* Criterio de Aceptación: La aplicación permite añadir nuevas funcionalidades y crecer en volumen de datos sin modificaciones estructurales.

## Especificación de requisitos

### Casos de uso

Diagrama

Descripción generada automáticamente

Figura 8 Diagrama de casos de uso

|  |  |
| --- | --- |
| CU-1 | Importación de requisitos |
| Versión | 1.0 |
| Autor | Estíbalitz Díez |
| Requisitos asociados | RF1, RF1.1, RF1.2, RF1.3, RF1.4 |
| Descripción | Permite al usuario cargar un documento PDF y asociarlo a un proyecto específico en el sistema de gestión de requisitos. |
| Precondición | El usuario ha iniciado sesión en el sistema y tiene permisos para cargar documentos. |
| Acciones | 1. El usuario selecciona la opción para cargar un documento PDF.  2. Muestra interfaz para seleccionar el archivo PDF desde su equipo.  3. El sistema permite asociar el PDF a un proyecto existente o crear uno nuevo.  4. El sistema extrae automáticamente la estructura de capítulos y requisitos del PDF.  5. El usuario revisa y edita los requisitos extraídos si es necesario.  6. El sistema almacena el documento, los requisitos y sus asociaciones en la base de datos. |
| Postcondición | El documento PDF y los requisitos extraídos están almacenados en la base de datos, asociados al proyecto seleccionado. |
| Excepciones | Error al cargar el archivo PDF (se muestra mensaje de error y se solicita intentar nuevamente). |
| Importancia | Alta |

Tabla 2 Cargar Requisitos – CU 1- Importación de requisitos

|  |  |
| --- | --- |
| CU-2 | Asociación de Documentos a Proyectos |
| Versión | 1.0 |
| Autor | Estíbalitz Díez |
| Requisitos asociados | RF1.1 |
| Descripción | Permite al usuario asociar un documento PDF a un proyecto existente o crear uno nuevo en el sistema. |
| Precondición | El usuario ha cargado un documento PDF en el sistema. |
| Acciones | 1. El usuario selecciona un proyecto existente o crea uno nuevo.  2. El sistema asigna el documento PDF al proyecto seleccionado. |
| Postcondición | El documento PDF está asociado a un proyecto específico. |
| Excepciones | Error al crear un proyecto nuevo o al seleccionar uno existente. |
| Importancia | Media |

Tabla 3 Cargar Requisitos – CU 2 – Asociación de proyectos

|  |  |
| --- | --- |
| CU-3 | Extracción de Capítulos y Requisitos |
| Versión | 1.0 |
| Autor | Estíbalitz Díez |
| Requisitos asociados | RF1.2 |
| Descripción | Permite la extracción automática de la estructura de capítulos y requisitos desde un documento PDF. |
| Precondición | El documento PDF ha sido cargado y asociado a un proyecto. |
| Acciones | 1. El sistema analiza el contenido del PDF.  2. Se identifica y extrae la estructura de capítulos y requisitos.  3. Los capítulos y requisitos extraídos son presentados al usuario para revisión. |
| Postcondición | La estructura de capítulos y requisitos está disponible en el sistema para revisión y edición. |
| Excepciones | Error en el proceso de extracción de capítulos y requisitos. |
| Importancia | Alta |

Tabla 4 Cargar Requisitos – CU 3 – Extracción de capítulos y requisitos

|  |  |
| --- | --- |
| CU-4 | Edición de Requisitos |
| Versión | 1.0 |
| Autor | Estíbalitz Díez |
| Requisitos asociados | RF1.3 |
| Descripción | Permite al usuario editar los requisitos extraídos del PDF antes de almacenarlos en la base de datos. |
| Precondición | Los requisitos han sido extraídos del documento PDF y están listos para revisión. |
| Acciones | 1. El usuario revisa los requisitos extraídos.  2. Realiza las modificaciones necesarias en el contenido de los requisitos.  3. Confirma los cambios para proceder con el almacenamiento. |
| Postcondición | Los requisitos editados están listos para ser almacenados en la base de datos. |
| Excepciones | Error al guardar las modificaciones en los requisitos. |
| Importancia | Alta | |

Tabla 5 Cargar Requisito – CU 4 – Edición de requisitos

|  |  |
| --- | --- |
| CU-5 | Almacenamiento de Datos en la Base de Datos |
| Versión | 1.0 |
| Autor | Estíbalitz Díez |
| Requisitos asociados | RF1.4 |
| Descripción | Permite almacenar el documento, los requisitos y sus asociaciones en la base de datos local del sistema. |
| Precondición | Los requisitos han sido revisados y confirmados por el usuario. |
| Acciones | 1. El sistema organiza los datos para el almacenamiento.  2. Se almacenan el documento, los requisitos y las asociaciones en la base de datos. |
| Postcondición | La información del documento y los requisitos están disponibles en la base de datos para futuras consultas y actualizaciones. |
| Excepciones | Error al guardar los datos en la base de datos. | |
| Importancia | Alta | |

Tabla 6 Cargar Requisitos – CU 5 – Almacenamiento en BBDD

|  |  |
| --- | --- |
| CU-6 | Validación de Asociaciones con Subsistemas |
| Versión | 1.0 |
| Autor | Estíbalitz Díez |
| Requisitos asociados | RF4, RF4.1 |
| Descripción | Permite al usuario validar y ajustar las asociaciones de subsistemas para cada requisito antes de guardarlos en la base de datos. |
| Precondición | El sistema ha sugerido asociaciones de subsistemas para cada requisito extraído. |
| Acciones | 1. El usuario revisa las asociaciones sugeridas.  2. Ajusta las asociaciones de subsistemas si es necesario.  3. Confirma la validación para proceder con el almacenamiento. |
| Postcondición | Los requisitos están validados con las asociaciones de subsistemas correspondientes. |
| Excepciones | Error al guardar las asociaciones de subsistemas. |
| Importancia | Alta |

Tabla 7 Cargar Requisitos – CU 6 – Validación de Asociaciones

|  |  |
| --- | --- |
| CU-7 | Consulta de Documentos y Requisitos |
| Versión | 1.0 |
| Autor | Estíbalitz Díez |
| Requisitos asociados | RF2, RF2.1, RF2.2 |
| Descripción | Permite al usuario realizar consultas de documentos y requisitos filtrando por proyecto, subsistema o fecha de creación. |
| Precondición | Existen documentos y requisitos almacenados en la base de datos. |
| Acciones | 1. El usuario selecciona los filtros de consulta (por proyecto, subsistema, etc.).  2. El sistema muestra los resultados basados en los filtros aplicados. |
| Postcondición | Se presenta una lista de documentos y requisitos que cumplen con los filtros de consulta. |
| Excepciones | No se encuentran resultados para los filtros seleccionados. |
| Importancia | Alta |

Tabla 8 Consulta CU 7 – Consulta de documentos y Requisitos

|  |  |
| --- | --- |
| CU-8 | Exportación de Resultados |
| Versión | 1.0 |
| Autor | Estíbalitz Díez |
| Requisitos asociados | RF3, RF3.1 |
| Descripción | Permite la exportación de los resultados de las consultas en un formato compatible, como CSV. |
| Precondición | Existen resultados de una consulta previa que el usuario desea exportar. |
| Acciones | 1. El usuario selecciona la opción de exportación.  2. Elige el formato de exportación (por ejemplo, CSV).  3. El sistema genera el archivo de exportación y lo pone a disposición del usuario. |
| Postcondición | Se genera un archivo de exportación con los resultados de la consulta. |
| Excepciones | Error al generar el archivo de exportación. |
| Importancia | Media |

Tabla 9 Consulta CU 8- Exportación de requisitos

|  |  |
| --- | --- |
| CU-9 | Actualización de BBDD |
| Versión | 1.0 |
| Autor | Estíbalitz Díez |
| Requisitos asociados | RF1.5, RF1.6 |
| Descripción | Permite al usuario añadir / eliminar proyectos, subsistemas y relaciones almacenados en la base de datos. |
| Precondición | La BBDD ha sido creada. |
| Acciones | 1. El accede a proyectos / subsistemas / asociaciones. .  2. Realiza la acción deseada.  3. Visualiza la modificación realizada. |
| Postcondición | La base de datos ha sido actualizada. |
| Excepciones | Error al guardar la modificación. |
| Importancia | Alta | |

Tabla 10 Actualización BBDD CU 9 - Actualización

|  |  |
| --- | --- |
| CU-10 | Manejo Multilingüe |
| Versión | 1.0 |
| Autor | Estíbalitz Díez |
| Requisitos asociados | RF5, RF5.1 |
| Descripción | Permite al usuario seleccionar el idioma de la interfaz para facilitar el uso en entornos internacionales. |
| Precondición | La aplicación está configurada con soporte multilingüe. |
| Acciones | 1. El usuario selecciona el idioma deseado.  2. La interfaz cambia al idioma seleccionado para todas las funcionalidades disponibles. |
| Postcondición | La interfaz se muestra en el idioma seleccionado. |
| Excepciones | Error al cambiar el idioma de la interfaz. |
| Importancia | Media |

Tabla 11 Idioma CU 10 – Manejo Multilingüe

# Especificación de diseño

## Introducción al diseño

En este capítulo incluimos el diseño de los datos, detallando la estructura, el diagrama E/R y el diagrama relacional, también el diseño procedimental mediante los diagramas de secuencias y el diseño arquitectónico.

## Diseño de datos

En este proyecto de escritorio para los datos principales de la aplicación se maneja una base de datos con la que se interactúa desde el software de la aplicación para la generación de nuevo registros, consultas, creación de nuevas relaciones y eliminaciones. Esta estructura es la que da soporte a la información relativa a los documentos sobre los que se realiza la extracción.

En este proyecto, manejamos otros datos a lo que nos referiremos como secundarios ya que no son propios de la gestión de los requisitos, pero son necesarios. Son los relativos al manejo de los distintos idiomas del programa y a los que definimos como Tokenes, los relativos a las palabras clave que tienen como función la detección del alcance para los subsistemas. Para la primera versión del software, para los datos secundarios de los idiomas se ha optado por archivos JSON, y para los datos correspondientes a los Tokenes se usan archivos CSV.

### Estructura de datos

La estructura de datos de la aplicación está diseñada para organizar y gestionar la información de los requisitos contenidos en los documentos que tienen alcance para los distintos subsistemas en cada proyecto de ingeniería.

A continuación, se describe cada entidad principal y sus relaciones.

#### Modelos de Datos

1. Proyectos:

* **Descripción**: Esta tabla almacena los proyectos (dando nombre al mismo por la ciudad donde se vaya a realizar) sobre los que se vayan a gestionar los requisitos. (Se pueden gestionar proyectos a los que se vaya a licitar con requisitos específicos de oferta y/o requisitos de proyectos ya adjudicados).
* **Atributos**:
* **id (PK):** Identificador único del proyecto.
* **nombre:** Nombre de la ciudad. Este campo es único y no permite valores nulos.
* **Relación:** Un proyecto puede estar asociada a varios documentos, pero cada documento pertenece a un único proyecto.

1. Subsistemas:

* **Descripción**: Almacena los distintos subsistemas que pueden estar vinculados a los documentos.
* **Atributos**:
* **id (PK):** Identificador único del subsistema.
* **nombre:** Nombre del subsistema.
* **Relación**: Un subsistema puede estar asociado a múltiples documentos, y un documento puede relacionarse con múltiples subsistemas.

1. Documentos:

* **Descripción**: Representa un documento dentro del sistema, que contiene una versión específica y está asociado a un proyecto.
* **Atributos**:
* **id (PK):** Identificador único del documento.
* **título:** Título del documento.
* **versión:** Versión del documento.
* **fecha:** Fecha de creación o modificación del documento (con valor predeterminado a la fecha actual).
* **proyecto\_id (FK):** Referencia al proyecto al que pertenece el documento.

**Relación**: Cada documento está asociado a un único proyecto y puede estar relacionado con múltiples subsistemas.

1. Requisitos:

* **Descripción:** Almacena los requisitos asociados a cada documento, permitiendo registrar el contenido de cada requisito junto con su capítulo correspondiente.
* **Atributos:**
* **id (PK):** Identificador único del requisito.
* **capitulo:** Número de capítulo en el que se encuentra el requisito.
* **requisito:** Descripción del requisito.
* **documento\_id (FK):** Referencia al documento al que pertenece el requisito.
* **Relación:** Un documento puede tener múltiples requisitos asociados, pero cada requisito está vinculado a un único documento.

1. Asociación entre Documentos y Subsistemas:

* **Descripción:** Esta tabla intermedia permite gestionar la relación muchos-a-muchos entre los documentos y los subsistemas.
* **Atributos:**
* **documento\_id (FK):** Referencia al documento.
* **subsistema\_id (FK):** Referencia al subsistema.
* **Relación:** Un documento puede estar vinculado a varios subsistemas y un subsistema puede estar presente en varios documentos.

#### Descripción de Relaciones

* **Relación entre Proyectos y Documentos:** Es una relación uno a muchos, donde un proyecto puede tener varios documentos, pero cada documento pertenece a un único proyecto.
* **Relación entre Documentos y Subsistemas:** Es una relación muchos a muchos gestionada a través de la tabla intermedia "Asociación entre Documentos y Subsistemas".
* **Relación entre Documentos y Requisitos:** Es una relación uno a muchos, donde un documento puede tener múltiples requisitos asociados, pero cada requisito está vinculado a un único documento.

Este diseño permite una organización clara y escalable de la información, facilitando la trazabilidad y la gestión de requisitos en proyectos complejos.

### Diagrama E/R

Diagrama

Descripción generada automáticamente

Figura 9 Diagrama Entidad - Relación

### Diagrama Relacional

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente con confianza media

Figura 10 Diagrama relacional

## Diseño procedimental

### Diagrama secuencias Cargar

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Figura 11 Diagrama Secuencia CARGAR

### Diagrama secuencias Consulta

Diagrama

Descripción generada automáticamente

Figura 12 Diagrama Secuencia CONSULTA

### Diagrama secuencia Acciones

Word

Descripción generada automáticamente

Figura 13 Diagrama Secuencia Otras acciones

## Diseño arquitectónico

En esta sección detallamos la arquitectura general, los componentes principales, el flujo de los datos, los patrones de diseño y del repositorio.

### Descripción General de la Arquitectura

La aplicación sigue una arquitectura en capas específica para aplicaciones de escritorio en Python. Los componentes principales incluyen:

Interfaz de Usuario: Construida con tkinter, proporciona una GUI que permite a los usuarios cargar documentos PDF, verificar asociaciones y consultar los requisitos.

Lógica de Negocio: Implementada en Python, esta capa maneja el procesamiento del documento, la extracción de información, el análisis de palabras clave y la asociación con subsistemas.

Base de Datos: Utiliza sqlite3 para el almacenamiento de datos estructurados como documentos, requisitos y asociaciones.

### Componentes Principales

Interfaz de Usuario (UI): Permite al usuario interactuar con la aplicación a través de ventanas de tkinter, donde puede cargar documentos y realizar consultas.

Procesamiento de Documento: Esta funcionalidad extrae texto de los PDFs, organiza la información en capítulos y requisitos, y busca palabras clave para sugerir asociaciones.

Base de Datos (sqlite3): Gestiona el almacenamiento y la recuperación de datos, facilitando la persistencia de documentos, requisitos y relaciones de subsistemas.

### Flujo de Datos

Carga de Documentos: La interfaz de usuario envía el documento a la capa de procesamiento, donde se extrae la información y se almacenan los requisitos y asociaciones en la base de datos.

Consulta de Información: El usuario puede realizar consultas en la base de datos desde la interfaz, y los resultados se muestran en la misma, permitiendo explorar documentos, requisitos y subsistemas asociados.

### Patrones de Diseño Utilizados

Patrón de Capa: Organiza la arquitectura en diferentes capas (UI, lógica de negocio y base de datos), asegurando que cada una cumple un rol específico.

### Patrón Repository (simplificado):

La interacción con sqlite3 se encapsula, proporcionando una capa de acceso a la base de datos que simplifica la manipulación de datos sin exponer detalles internos.

#### Diseño del directorio.

¿

Único punto pendiente

# Documentación técnica de programación

## Introducción

## Estructura de directorios

## Manual del programador

## Pruebas del sistema

# Documentación de usuario

## Introducción

## Requisitos de usuario

## Instalación

## Manual del usuario

I

# Anexo de sostenibilización curricular

## Introducción

## Competencias en sostenibilidad adquiridas

## Impacto social y ambiental

Este proyecto de desarrollo de una aplicación de gestión de requisitos de ingeniería promueve la sostenibilidad en varios aspectos clave. A través de la digitalización y el procesamiento automatizado de documentos PDF, esta herramienta facilita una gestión de requisitos más eficiente, lo que contribuye a la reducción del uso de papel y otros recursos físicos, promoviendo prácticas más sostenibles en el entorno de la ingeniería.

Sostenibilidad Ambiental

La aplicación permite el almacenamiento digital y la organización de documentos de requisitos, eliminando la necesidad de imprimir o archivar documentos en formato físico. Al reducir el consumo de papel y tinta, el proyecto contribuye a disminuir el impacto ambiental asociado a la impresión y almacenamiento de documentos. Además, al contar con una arquitectura escalable y una base de datos que permite la exportación de datos en formato CSV, la herramienta facilita el manejo eficiente de la información sin necesidad de producir copias adicionales en papel.

Sostenibilidad Económica

Desde el punto de vista económico, el uso de tecnologías de código abierto como Python y SQLite minimiza los costos de desarrollo y mantenimiento de la aplicación, permitiendo que las organizaciones puedan implementar esta solución sin incurrir en gastos elevados. Asimismo, la capacidad de exportación a otros sistemas de gestión de requisitos asegura que la información recopilada y organizada pueda integrarse con herramientas avanzadas, maximizando así el valor de los datos sin necesidad de inversiones adicionales en software.

Sostenibilidad Social

La aplicación está diseñada para ser intuitiva y fácil de usar, promoviendo el acceso inclusivo a herramientas de gestión de requisitos. Esto beneficia especialmente a pequeñas y medianas empresas o equipos que no pueden acceder a soluciones de software de gestión de requisitos de alto costo. Al facilitar la organización y consulta de información en múltiples idiomas, la aplicación también garantiza su accesibilidad en diferentes contextos lingüísticos y culturales.

Sostenibilidad Técnica y Escalabilidad

Desde un punto de vista técnico, la arquitectura modular de la aplicación y el uso de archivos de configuración independientes permiten fácilmente la expansión y actualización del sistema. Esto asegura que la herramienta pueda adaptarse a nuevas necesidades y tecnologías, evitando la obsolescencia y reduciendo la necesidad de desarrollar nuevas aplicaciones en el futuro. La capacidad de integrar más palabras clave, idiomas y configuraciones específicas asegura una larga vida útil del sistema y un crecimiento sostenible en el tiempo.

Impacto Ético

El desarrollo y uso de la aplicación se ha orientado hacia prácticas responsables y éticas, promoviendo un uso eficiente de los recursos y un acceso equitativo a las herramientas de gestión. Al ser una aplicación de código abierto, cualquier organización puede implementar esta solución sin limitaciones de acceso, promoviendo así la democratización de tecnologías de gestión

Icono

Descripción generada automáticamente

Figura F.3.1 Imagen ejemplo

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Tabla 12 Tabla modelo

Bibliografía

https://lucid.app/documents#/documents?folder\_id=home