|  |  |  |
| --- | --- | --- |
| 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](#_Toc181727242)

[Índice de figuras V](#_Toc181727243)

[Índices de tablas VI](#_Toc181727244)

[*Apéndice A-* Plan de proyecto Software 1](#_Toc181727245)

[A.1. Introducción 1](#_Toc181727246)

[A.2. Planificación temporal 1](#_Toc181727247)

[A.3. Estudio de viabilidad 2](#_Toc181727248)

[*Apéndice B-* Especificación de Requisitos 4](#_Toc181727249)

[B.1. Introducción 5](#_Toc181727250)

[B.2. Objetivos generales 5](#_Toc181727251)

[B.3. Catálogo de requisitos 6](#_Toc181727252)

[B.3.1 Requisitos Funcionales 6](#_Toc181727253)

[B.3.2 Requisitos No funcionales 9](#_Toc181727254)

[B.4. Especificación de requisitos 10](#_Toc181727255)

[*Apéndice C-* Especificación de diseño 11](#_Toc181727256)

[C.1. Introducción al diseño 12](#_Toc181727257)

[C.2. Diseño de datos 12](#_Toc181727258)

[C.2.1 Estructura de datos 12](#_Toc181727259)

[C.2.2 Diagrama E/R 14](#_Toc181727260)

[C.2.3 Diagrama Relacional 15](#_Toc181727261)

[C.3. Diseño procedimental 16](#_Toc181727262)

[C.3.1 Diagrama secuencias Cargar 16](#_Toc181727263)

[C.3.2 Diagrama secuencias Consulta 17](#_Toc181727264)

[C.3.3 Diagrama secuencia Acciones 18](#_Toc181727265)

[C.4. Diseño arquitectónico 19](#_Toc181727266)

[C.4.1 Descripción General de la Arquitectura 19](#_Toc181727267)

[C.4.2 Componentes Principales 19](#_Toc181727268)

[C.4.3 Flujo de Datos 19](#_Toc181727269)

[C.4.4 Patrones de Diseño Utilizados 19](#_Toc181727270)

[C.4.5 Patrón Repository (simplificado): 19](#_Toc181727271)

[*Apéndice D-* Documentación técnica de programación 20](#_Toc181727272)

[D.1. Introducción 21](#_Toc181727273)

[D.2. Estructura de directorios 21](#_Toc181727274)

[D.3. Manual del programador 21](#_Toc181727275)

[D.4. Pruebas del sistema 21](#_Toc181727276)

[*Apéndice E-* Documentación de usuario 22](#_Toc181727277)

[E.1. Introducción 23](#_Toc181727278)

[E.2. Requisitos de usuario 23](#_Toc181727279)

[E.3. Instalación 23](#_Toc181727280)

[E.4. Manual del usuario 23](#_Toc181727281)

[*Apéndice F-* Anexo de sostenibilización curricular 24](#_Toc181727282)

[F.1. Introducción 25](#_Toc181727283)

[F.2. Competencias en sostenibilidad adquiridas 25](#_Toc181727284)

[F.3. Impacto social y ambiental 25](#_Toc181727285)

[Bibliografía 27](#_Toc181727286)

Índice de figuras

[Figura 1 Diagrama Entidad - Relación 14](#_Toc181727367)

[Figura 2 Diagrama relacional 15](#_Toc181727368)

[Figura 3 Diagrama Secuencia CARGAR 16](#_Toc181727369)

[Figura 4 Diagrama Secuencia CONSULTA 17](#_Toc181727370)

[Figura 5 Diagrama Secuencia Otras acciones 18](#_Toc181727371)

[Figura F.3.1 Imagen ejemplo 26](#_Toc181727372)

Índices de tablas

[Tabla 1 Requisitos Funcionales 6](#_Toc181727474)

[Tabla 2 Tabla modelo 26](#_Toc181727475)

# 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.

El desarrollo del proyecto se ha estructurado en varias fases que incluyen la definición de requisitos, diseño de arquitectura, desarrollo, integración y pruebas. Cada una de estas fases está planificada para asegurar una ejecución eficiente del proyecto, minimizando riesgos y garantizando que el sistema final cumpla con las expectativas y requisitos establecidos.

## Planificación temporal

La Planificación Temporal de este proyecto ha sido diseñada para organizar y distribuir las tareas de desarrollo en un calendario claro y estructurado, garantizando que cada fase se complete dentro de los plazos establecidos. La planificación temporal se ha dividido en cinco fases principales, detalladas a continuación:

Definición de Requisitos (Semanas 1-2): En esta fase inicial, se identifican y documentan los requisitos funcionales y no funcionales de la aplicación. Incluye la especificación de características como la carga de documentos PDF, el análisis de texto para identificación de subsistemas, la exportación de datos y la configuración para soporte multilingüe.

Diseño de Arquitectura (Semanas 3-4): Se desarrolla la estructura general de la aplicación, definiendo la organización de módulos y componentes. En esta fase se establece la modularidad del sistema, separando la lógica de negocio (procesamiento de PDF, análisis de palabras clave) de la interfaz de usuario y el acceso a la base de datos SQLite, asegurando así la escalabilidad y mantenibilidad.

Desarrollo y Codificación (Semanas 5-8): Durante esta fase, se implementa la funcionalidad principal de la aplicación. Las actividades incluyen la programación de la carga y análisis de documentos PDF, la asociación de requisitos a subsistemas, el manejo de palabras clave desde archivos externos, y la configuración de la exportación de datos en CSV.

Integración y Pruebas (Semanas 9-10): Se realizan pruebas de funcionamiento y se verifica que el sistema cumpla con los requisitos definidos. Esta fase asegura que todas las funciones, desde la carga y extracción de datos hasta la exportación y la compatibilidad multiplataforma, se integren correctamente y operen sin problemas.

Documentación y Preparación para Entrega (Semana 11): En esta última fase, se elabora la documentación de usuario, incluyendo la guía de uso y la ayuda integrada en los distintos idiomas soportados. Se preparan los archivos de configuración para facilitar futuras expansiones de la base de datos, la adición de nuevas palabras clave y la incorporación de idiomas adicionales.

Esta planificación temporal ha sido diseñada para facilitar el control de avance del proyecto, permitiendo ajustar los tiempos en caso de que surjan contratiempos y asegurando la entrega del sistema final en el plazo previsto.

Sprintssssssss

## 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 | 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. |
|  | RF1.4 | Actualización de Documentos y Requisitos | El sistema debe permitir la actualización de la información de los documentos y requisitos previamente almacenados. |
|  | RG1.5 | 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. |
| RF3 | | EXPORTACIÓN DE RESULTADOS DE CONSULTA | El sistema debe permitir la exportación de los resultados de las consultas en formato PDF o Excel. |
| RF4 | | ANÁLISIS DE TEXTO PARA IDENTIFICAR SUBSISTEMAS | 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. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabla Requisitos Funcionales

* **RF1. Carga de Documentos PDF**

Descripción: El sistema debe permitir al usuario cargar un documento PDF y asociarlo a un proyecto específico.

Criterio de Aceptación: El sistema confirma la carga y muestra el nombre del proyecto al cual está asociado el documento.

* RF1.1. Asociación de Documentos a Proyectos

Descripción: El sistema debe permitir seleccionar un proyecto existente o crear uno nuevo al momento de asociar un documento PDF.

Criterio de Aceptación: El proyecto seleccionado o creado aparece vinculado al documento cargado.

* RF1.2. Extracción de Capítulos y Requisitos

Descripción: El sistema debe extraer automáticamente la estructura de capítulos y requisitos desde el documento PDF.

Criterio de Aceptación: Los capítulos y requisitos extraídos se muestran en una lista editable.

* RF1.3. Almacenamiento de Datos en la Base de Datos

Descripción: El sistema debe almacenar el documento, los requisitos y sus asociaciones validadas en una base de datos local.

Criterio de Aceptación: La información se guarda de forma persistente y es recuperable en consultas posteriores.

* RF1.4. Actualización de Documentos y Requisitos

Descripción: El sistema debe permitir la actualización de la información de los documentos y requisitos previamente almacenados.

Criterio de Aceptación: Los cambios realizados en documentos y requisitos se reflejan inmediatamente en la base de datos.

* RF1.5. Eliminación de Documentos y Asociaciones

Descripción: El sistema debe permitir la eliminación de documentos, junto con sus requisitos y asociaciones con subsistemas.

Criterio de Aceptación: Al eliminar un documento, se eliminan también sus requisitos y asociaciones, sin dejar datos residuales.

* **RF2. Consulta de Documentos y Requisitos**

Descripción: El sistema debe permitir al usuario realizar consultas de documentos y requisitos filtrando por proyecto, subsistema o fecha de creación.

Criterio de Aceptación: Los resultados de la consulta se muestran en una vista ordenada y son exportables.

* **RF3. Exportación de Resultados de Consulta**

Descripción: El sistema debe permitir la exportación de los resultados de las consultas en formato PDF o Excel.

Criterio de Aceptación: Los resultados de la consulta se exportan correctamente en el formato seleccionado por el usuario.

* **RF4. Análisis de Texto para Identificar Subsistemas**

Descripción: El sistema debe analizar cada capítulo y requisito buscando palabras clave que identifiquen los subsistemas relevantes.

Criterio de Aceptación: El sistema muestra una lista de subsistemas sugeridos en función de las palabras clave encontradas en el texto.

* RF4.1. Validación de Asociaciones con Subsistemas

Descripción: El sistema debe permitir al usuario validar y ajustar las asociaciones de subsistemas para cada requisito antes de guardar los datos.

Criterio de Aceptación: El usuario puede aceptar, modificar o eliminar asociaciones antes de que se guarden en la base de datos.

### 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

# 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 Diagrama Entidad - Relación

### Diagrama Relacional

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente con confianza media

Figura Diagrama relacional

## Diseño procedimental

### Diagrama secuencias Cargar

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Figura Diagrama Secuencia CARGAR

### Diagrama secuencias Consulta

Diagrama

Descripción generada automáticamente

Figura Diagrama Secuencia CONSULTA

### Diagrama secuencia Acciones

Word

Descripción generada automáticamente

Figura 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.

¿

# 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 . Imagen ejemplo

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

Tabla Tabla modelo

Bibliografía

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