

# Especificación de Requisitos del Software (ERS)

## FocusFlow TUI - Gestor de Tareas Avanzado

Versión: 1.0

Fecha: 2025-05-17

### Tabla de Contenidos

1. Introducción
  - 1.1. Propósito del Documento
  - 1.2. Alcance del Producto
  - 1.3. Definiciones, Acrónimos y Abreviaturas
  - 1.4. Vista General del Documento
2. Descripción General
  - 2.1. Visión del Producto
  - 2.2. Funciones del Producto
  - 2.3. Público Objetivo
  - 2.4. Restricciones Generales
  - 2.5. Suposiciones y Dependencias
3. Requisitos Específicos
  - 3.1. Requisitos Funcionales
  - 3.2. Requisitos de Interfaz de Usuario (TUI)
  - 3.3. Requisitos de Almacenamiento de Datos
  - 3.4. Requisitos No Funcionales
4. [Tecnologías Propuestas](#)
5. [Plan de Desarrollo \(Hoja de Ruta Simplificada\)](#)

## 1. Introducción

### 1.1. Propósito del Documento

Este documento define los requisitos funcionales y no funcionales para "FocusFlow TUI", una aplicación de gestión de tareas con interfaz de usuario en la terminal. Su objetivo es servir como guía para el diseño, desarrollo y pruebas de la aplicación, asegurando que el producto final cumpla con las expectativas y necesidades identificadas.

### 1.2. Alcance del Producto

FocusFlow TUI será una aplicación de línea de comandos para la gestión personal de tareas y proyectos pequeños. Permitirá a los usuarios crear, organizar, priorizar y dar seguimiento a sus pendientes de manera eficiente directamente desde la terminal. Se diferenciará por integrar herramientas de productividad como un temporizador

Pomodoro, vistas de calendario, y un sistema de "Modo Foco" para minimizar distracciones, además de ofrecer una experiencia de usuario TUI pulida y personalizable.

### 1.3. Definiciones, Acrónimos y Abreviaturas

- **TUI:** Text-based User Interface (Interfaz de Usuario Basada en Texto).
- **CRUD:** Create, Read, Update, Delete (Crear, Leer, Actualizar, Eliminar).
- **ERS:** Especificación de Requisitos del Software.
- **MVP:** Minimum Viable Product (Producto Mínimo Viable).
- **JSON:** JavaScript Object Notation.
- **CSV:** Comma-Separated Values.
- **SQLite:** Base de datos SQL ligera y autocontenida.

### 1.4. Vista General del Documento

Este documento se organiza en las siguientes secciones:

- **Sección 2 (Descripción General):** Ofrece una perspectiva general del producto, sus funciones principales, el público objetivo y las restricciones.
- **Sección 3 (Requisitos Específicos):** Detalla los requisitos funcionales, de interfaz, de datos y no funcionales.
- **Sección 4 (Tecnologías Propuestas):** Enumera las tecnologías que se consideran para el desarrollo.
- **Sección 5 (Plan de Desarrollo):** Esboza una posible hoja de ruta para la construcción del producto.

## 2. Descripción General

### 2.1. Visión del Producto

FocusFlow TUI aspira a ser más que un simple gestor de tareas en la terminal. Busca ser una herramienta integral de productividad para usuarios que prefieren la eficiencia y la personalización de la línea de comandos. Su diseño se centrará en minimizar la fricción en la gestión de tareas y en fomentar la concentración a través de características innovadoras, manteniendo una interfaz limpia, intuitiva y estéticamente agradable dentro de las limitaciones de una TUI.

#### Características Destacadas (Novedosas):

- **Integración Pomodoro Nativa:** Temporizador Pomodoro directamente vinculado a las tareas.
- **Vista de "Hoy" Inteligente:** Priorización automática y sugerencias para el día.
- **Modo Foco:** Interfaz minimalista que muestra solo la tarea actual y el

temporizador.

- **Mini-Calendario Integrado:** Visualización de tareas con fechas límite en un calendario compacto.
- **Etiquetas y Proyectos Flexibles:** Organización robusta con búsqueda y filtrado potentes.
- **Informes de Productividad Simples:** Resumen de tareas completadas y tiempo enfocado.

## 2.2. Funciones del Producto

Las funciones principales de FocusFlow TUI incluirán:

- Gestión completa de tareas (crear, ver, editar, eliminar, marcar como completada/incompleta).
- Organización de tareas en proyectos o listas.
- Asignación de prioridades y fechas de vencimiento.
- Etiquetado de tareas para categorización.
- Búsqueda y filtrado de tareas.
- Temporizador Pomodoro.
- Vista de calendario.
- Modo Foco.
- Configuración de la aplicación.
- Importación/Exportación de datos.

## 2.3. Público Objetivo

- Desarrolladores de software, administradores de sistemas y otros profesionales de TI que pasan mucho tiempo en la terminal.
- Usuarios que valoran la eficiencia, la personalización y las herramientas sin distracciones.
- Entusiastas de la línea de comandos que buscan alternativas a las aplicaciones GUI tradicionales.
- Personas interesadas en mejorar su productividad y gestión del tiempo.

## 2.4. Restricciones Generales

- La aplicación será desarrollada en Python.
- La interfaz será exclusivamente una TUI, compatible con terminales estándar (Linux, macOS, Windows con WSL o emuladores de terminal compatibles).
- El almacenamiento de datos será local (JSON, CSV o SQLite3). No se contempla sincronización en la nube para la versión inicial.
- El proyecto se desarrollará con un enfoque iterativo, comenzando con un MVP.

## 2.5. Suposiciones y Dependencias

- Los usuarios tendrán Python instalado en sus sistemas.
- Los usuarios poseen un conocimiento básico sobre cómo operar en una interfaz de línea de comandos.
- Se utilizarán bibliotecas TUI de Python como Textual (preferida por su modernidad) o Urwid.

### **3. Requisitos Específicos**

#### **3.1. Requisitos Funcionales**

##### **3.1.1. Gestión de Tareas (RF-001)**

- **RF-001.1:** El sistema permitirá crear nuevas tareas con un título descriptivo.
- **RF-001.2:** El sistema permitirá añadir una descripción detallada opcional a cada tarea.
- **RF-001.3:** El sistema permitirá asignar una prioridad a cada tarea (ej: Alta, Media, Baja).
- **RF-001.4:** El sistema permitirá asignar una fecha y hora de vencimiento opcionales a cada tarea.
- **RF-001.5:** El sistema permitirá marcar tareas como completadas o pendientes.
- **RF-001.6:** El sistema permitirá editar todos los atributos de una tarea existente.
- **RF-001.7:** El sistema permitirá eliminar tareas.
- **RF-001.8:** El sistema permitirá crear subtareas dependientes de una tarea principal (opcional, para versiones futuras).
- **RF-001.9:** El sistema permitirá establecer tareas recurrentes (ej: diaria, semanal, mensual).

##### **3.1.2. Gestión de Proyectos/Listas (RF-002)**

- **RF-002.1:** El sistema permitirá agrupar tareas en proyectos o listas personalizadas.
- **RF-002.2:** El sistema permitirá crear, renombrar y eliminar proyectos/listas.
- **RF-002.3:** El sistema permitirá mover tareas entre diferentes proyectos/listas.

##### **3.1.3. Sistema de Etiquetado (RF-003)**

- **RF-003.1:** El sistema permitirá asignar múltiples etiquetas (tags) a las tareas.
- **RF-003.2:** El sistema permitirá crear y gestionar etiquetas.
- **RF-003.3:** (Novedoso) El sistema podrá sugerir etiquetas basadas en el contenido de la tarea o etiquetas usadas frecuentemente en contextos similares.

##### **3.1.4. Vista de "Hoy" y Próximas Tareas (RF-004)**

- **RF-004.1:** El sistema ofrecerá una vista dedicada ("Hoy") que muestre las tareas programadas para el día actual y las tareas vencidas.

- **RF-004.2:** (Novedoso) La vista "Hoy" podrá tener una sección de "sugerencias" o tareas priorizadas automáticamente basadas en urgencia e importancia.
- **RF-004.3:** El sistema permitirá visualizar tareas programadas para los próximos 7 días.

### **3.1.5. Mini-Calendario Integrado (RF-005)**

- **RF-005.1:** (Novedoso) El sistema mostrará un mini-calendario (mensual o semanal) en la interfaz.
- **RF-005.2:** Los días con tareas programadas se resaltarán en el calendario.
- **RF-005.3:** Al seleccionar un día en el calendario, se mostrarán las tareas correspondientes a esa fecha.

### **3.1.6. Temporizador Pomodoro Integrado (RF-006)**

- **RF-006.1:** (Novedoso) El sistema incluirá un temporizador Pomodoro configurable (duración de trabajo, descanso corto, descanso largo, número de pomodoros por ciclo).
- **RF-006.2:** El sistema permitirá asociar una sesión Pomodoro a una tarea específica.
- **RF-006.3:** El sistema emitirá notificaciones (visuales en la TUI, y opcionalmente sonoras si es posible sin dependencias complejas) al finalizar cada intervalo.
- **RF-006.4:** El sistema registrará los pomodoros completados por tarea y por día.

### **3.1.7. Búsqueda y Filtrado (RF-007)**

- **RF-007.1:** El sistema permitirá buscar tareas por título, descripción y etiquetas.
- **RF-007.2:** El sistema permitirá filtrar tareas por proyecto/lista, prioridad, estado (pendiente/completada), fecha de vencimiento.
- **RF-007.3:** (Novedoso) Se explorará una búsqueda simple con lenguaje natural (ej: "tareas @trabajo para mañana", "reuniones #importante").

### **3.1.8. Modo Foco (RF-008)**

- **RF-008.1:** (Novedoso) El sistema ofrecerá un "Modo Foco" que oculte todos los elementos de la interfaz excepto la tarea actual seleccionada y el temporizador Pomodoro (si está activo).
- **RF-008.2:** Este modo buscará minimizar distracciones visuales.

### **3.1.9. Plantillas de Tareas/Proyectos (RF-009)**

- **RF-009.1:** (Novedoso) El sistema permitirá crear plantillas para tareas o listas de tareas recurrentes (ej: "checklist para nuevo proyecto", "rutina matutina").
- **RF-009.2:** El sistema permitirá instanciar nuevas tareas/proyectos a partir de estas plantillas.

### **3.1.10. Informes de Productividad (RF-010)**

- **RF-010.1:** (Novedoso) El sistema generará informes simples sobre tareas completadas (por día, semana, mes).
- **RF-010.2:** El sistema mostrará el tiempo total enfocado utilizando el temporizador Pomodoro.

### **3.1.11. Importación/Exportación de Datos (RF-011)**

- **RF-011.1:** El sistema permitirá exportar todas las tareas a formato JSON o CSV.
- **RF-011.2:** El sistema permitirá importar tareas desde formato JSON o CSV (con una estructura definida).

### **3.1.12. Configuración de Usuario (RF-012)**

- **RF-012.1:** El sistema permitirá configurar preferencias del usuario, como:
  - Formato de fecha/hora.
  - Duraciones del temporizador Pomodoro.
  - Paleta de colores de la TUI (si es soportado por la biblioteca TUI).
  - Ruta del archivo de datos.

## **3.2. Requisitos de Interfaz de Usuario (TUI)**

- **RUI-001:** La navegación principal se realizará mediante el teclado (flechas, Tab, Enter, Esc).
- **RUI-002:** Se definirán atajos de teclado intuitivos para las acciones más comunes (ej: n para nueva tarea, e para editar, d para eliminar, c para completar, s para buscar, f para modo foco).
- **RUI-003:** La interfaz proporcionará retroalimentación clara al usuario sobre las acciones realizadas (mensajes de confirmación, errores, estado).
- **RUI-004:** El diseño será limpio, organizado y legible, utilizando de manera efectiva el espacio limitado de la terminal.
- **RUI-005:** Se considerará el uso de colores y estilos de texto (negrita, subrayado) para mejorar la jerarquía visual y la usabilidad, de forma personalizable si es posible.
- **RUI-006:** La aplicación será responsiva a diferentes tamaños de ventana de terminal, ajustando su layout dinámicamente.
- **RUI-007:** Se mostrará ayuda contextual o una pantalla de ayuda accesible con los principales comandos.

## **3.3. Requisitos de Almacenamiento de Datos**

- **RDA-001:** Los datos de las tareas y la configuración del usuario se almacenarán localmente de forma persistente.
- **RDA-002:** Se evaluará y seleccionará un formato de almacenamiento:

- **JSON:** Simple de implementar, legible por humanos.
- **CSV:** Adecuado para datos tabulares simples, pero limitado para estructuras complejas.
- **SQLite3:** Más robusto para consultas complejas, integridad de datos y escalabilidad futura, aunque con una ligera mayor complejidad inicial. Se recomienda SQLite3 para las funcionalidades avanzadas planeadas.
- **RDA-003:** El sistema asegurará la integridad de los datos, evitando corrupción en la medida de lo posible.
- **RDA-004:** El archivo de datos será accesible y gestionable por el usuario (ej: para copias de seguridad manuales).

### 3.4. Requisitos No Funcionales

- **RNF-001 (Usabilidad):** La aplicación será fácil de aprender y usar para el público objetivo. La navegación y las operaciones comunes serán intuitivas.
- **RNF-002 (Rendimiento):** La aplicación responderá rápidamente a las interacciones del usuario. Las operaciones de carga, guardado y búsqueda no deben generar retrasos notables, incluso con un número considerable de tareas (ej: varios miles).
- **RNF-003 (Fiabilidad):** La aplicación será estable y no perderá datos del usuario. Se manejarán adecuadamente los errores.
- **RNF-004 (Portabilidad):** Siendo una aplicación Python TUI, deberá funcionar en los principales sistemas operativos donde Python y una terminal compatible estén disponibles (Linux, macOS, Windows con WSL).
- **RNF-005 (Mantenibilidad):** El código estará bien estructurado, comentado y modularizado para facilitar futuras actualizaciones y correcciones.
- **RNF-006 (Personalización):** Se permitirá un grado razonable de personalización de la apariencia (colores) y comportamiento (atajos, configuración Pomodoro).

## 4. Tecnologías Propuestas

- **Lenguaje de Programación:** Python (versión 3.8 o superior).
- **Bibliotecas TUI:**
  - **Textual (Preferida):** Framework moderno para TUIs ricas e interactivas, basado en Rich. Ofrece componentes avanzados y un sistema de layout similar a CSS.
  - **Urwid:** Alternativa robusta con un conjunto completo de widgets y buen manejo de eventos.
- **Almacenamiento de Datos:**
  - **SQLite3 (Recomendado):** Para una gestión de datos más estructurada y eficiente, especialmente con las funcionalidades de búsqueda, filtrado y

relaciones entre tareas/proyectos.

- **JSON:** Opción más simple si se prefiere un archivo plano único, aunque puede volverse menos eficiente con grandes cantidades de datos o relaciones complejas.

- **Otras Bibliotecas Potenciales:**

- datetime, time (para manejo de fechas y temporizador).
- uuid (para identificadores únicos de tareas).
- configparser (para configuraciones de usuario si no se integran en la base de datos).

## 5. Plan de Desarrollo (Hoja de Ruta Simplificada)

Se propone un desarrollo iterativo e incremental:

- **Fase 1: Producto Mínimo Viable (MVP)**

- Funcionalidades básicas de gestión de tareas (CRUD, prioridades, fechas de vencimiento).
- Gestión básica de proyectos/listas.
- Almacenamiento de datos (elegir formato e implementar).
- Interfaz TUI básica navegable.
- **Característica Destacada Inicial:** Implementación del Temporizador Pomodoro integrado.

- **Fase 2: Funcionalidades Avanzadas y Experiencia de Usuario**

- Implementación del sistema de etiquetado.
- Desarrollo de la Vista "Hoy" y Próximas Tareas.
- Integración del Mini-Calendario.
- Implementación del Modo Foco.
- Mejoras en la búsqueda y filtrado.
- Configuración de usuario.
- Refinamiento de la TUI (estética, atajos, ayuda).

- **Fase 3: Características Adicionales y Pulido**

- Implementación de Plantillas de Tareas/Proyectos.
- Desarrollo de Informes de Productividad.
- Importación/Exportación de datos.
- Sugerencias inteligentes de etiquetas (si es viable).
- Optimización del rendimiento.
- Pruebas exhaustivas y corrección de errores.
- Documentación final para el usuario.

- **Fase 4 (Post-Lanzamiento/Futuro):**

- Subtareas.



- Tareas recurrentes avanzadas.
- Gamificación (puntos, rachas).
- Considerar opciones de sincronización (si hay demanda).

Este documento sirve como punto de partida. A medida que el proyecto avance, es posible que algunos requisitos se modifiquen o se añadan nuevos.