Análisis de Requisitos HLC

Sergio Puga Luque

Christian Hierro Cordón

Leonel Yupanqui Serrano

Pedro Cuadrado Moreno

INTRODUCCIÓN

- **Título del Proyecto**

  Aplicación de Gestión de Contraseñas con Interfaz Gráfica.

- **Propósito del Documento**

  Este documento tiene como objetivo establecer los requerimientos funcionales y no funcionales necesarios para desarrollar "PassSafe", un gestor de contraseñas seguro y fácil de usar. La aplicación está diseñada para permitir a los usuarios almacenar, organizar y gestionar sus contraseñas de forma eficiente, garantizando la seguridad y privacidad de sus datos.

  PassSafe busca resolver el problema de recordar contraseñas múltiples al ofrecer una plataforma centralizada donde los usuarios puedan administrar credenciales de diferentes servicios. Además, incluye características como generación de contraseñas aleatorias seguras.

1. Requisitos Funcionales

Estos son los requisitos que describen el comportamiento del sistema y las funcionalidades que debe proporcionar.

1.1. Autenticación del Usuario

- **Descripción**: El gestor de contraseñas debe permitir al usuario autenticar su identidad para acceder a sus contraseñas almacenadas mediante una contraseña maestra.
- **Funcionalidades**:
    - El usuario debe ingresar una contraseña maestra al inicio.
    - Si la contraseña es correcta, el acceso es permitido; si no, el acceso se deniega.

1.2. Almacenamiento de Contraseñas

- **Descripción**: Las contraseñas deben almacenarse en el equipo de forma local.
- **Funcionalidades**:
    - El sistema debe permitir almacenar las contraseñas en un archivo.

1.3. Añadir nueva contraseña

- **Descripción**: El usuario debe poder agregar nuevas contraseñas de manera sencilla.
- **Funcionalidades**:
    - El usuario puede ingresar el nombre del servicio (por ejemplo, "Google") y la contraseña correspondiente.

1.4. Ver Contraseñas Almacenadas

- **Descripción**: El gestor debe permitir al usuario ver las contraseñas almacenadas
- **Funcionalidades**:
    - El usuario debe ser capaz de visualizar las contraseñas relacionadas con los diferentes servicios como Google por ejemplo, en el caso anterior.

1.5. Buscar contraseñas

- **Descripción**: El sistema debe permitir al usuario buscar rápidamente una contraseña de un servicio específico.
- **Funcionalidades**:
    - El usuario debe poder buscar por nombre del servicio (por ejemplo, "Facebook", "Twitter") y obtener la contraseña correspondiente de manera rápida.

1.6. Eliminar Contraseñas

- **Descripción**: El gestor debe permitir al usuario eliminar contraseñas almacenadas.
- **Funcionalidades**:
    - El usuario debe poder seleccionar una contraseña y eliminarla del sistema.
    - Las contraseñas eliminadas deben eliminarse de forma automática del archivo JSON.

1.7. Generar Contraseñas Aleatorias

- **Descripción**: El sistema debe ser capaz de generar contraseñas aleatorias y seguras para el usuario.
- **Funcionalidades**:
    - El usuario puede solicitar una contraseña generada automáticamente, que cumpla con los requisitos de seguridad (por ejemplo, incluir mayúsculas, minúsculas, números y símbolos).

2. Requisitos No Funcionales

Estos requisitos están relacionados con la calidad y rendimiento del sistema, no con las funcionalidades específicas.

2.1. Facilidad de Uso (Usabilidad)

- **Descripción**: La aplicación debe ser fácil de usar para que cualquier persona, incluso sin experiencia, pueda gestionar sus contraseñas sin problemas.
- **Requisitos**:
    - La interfaz de usuario debe ser clara y sencilla.
    - Los mensajes de error y las instrucciones deben ser comprensibles y amigables.
    - Las acciones más comunes (agregar, ver, buscar y eliminar contraseñas) deben ser accesibles.

2.2. Escalabilidad

- **Descripción**: El sistema debe poder manejar un número creciente de contraseñas sin perder rendimiento.
- **Requisitos**:
    - Al almacenar las contraseñas en archivos, el sistema debe ser capaz de gestionar grandes volúmenes de contraseñas sin afectar la velocidad de las operaciones.
    - Se usarán archivos JSON para almacenar las contraseñas guardadas.

2.3. Rendimiento

- **Descripción**: El gestor de contraseñas debe ser rápido y eficiente en sus operaciones.
- **Requisitos**:
    - Las búsquedas de contraseñas deben ser rápidas incluso cuando el número de contraseñas almacenadas sea grande.

3. Requisitos de Seguridad Específicos

3.1. Protección contra ataques de fuerza bruta

- **Descripción**: La contraseña maestra debe ser suficientemente segura para evitar ataques de fuerza bruta.
- **Requisitos**:
    - La contraseña maestra debe ser larga y compleja.
    - Se pueden implementar medidas como un número máximo de intentos fallidos para evitar intentos de adivinación.

4. Tecnologías y Herramientas Recomendadas

- **Lenguaje de programación**: **Python**
- **Archivo de texto plano: JSON**
- **Bibliotecas**:
    - **json** (para almacenamiento de contraseñas en formato JSON).
    - **getpass** (para entrada segura de contraseñas sin mostrarlas en la terminal).
    - **Tkinter** (para la interfaz gráfica de usuario si decides incluir una).

5. Resumen de Requisitos

- **Funcionales**:
    - Gestión de contraseñas (añadir, buscar, ver, eliminar).
    - Autenticación con una contraseña maestra.
    - Generación de contraseñas aleatorias seguras.
- **No Funcionales**:
    - **Seguridad**: Protección de claves, autenticación.
    - **Usabilidad**: Interfaz sencilla y amigable.
    - **Escalabilidad**: Capaz de manejar un número alto de contraseñas.
    - **Rendimiento**: Eficiente en cuanto a tiempo de respuesta.
- **Tecnologías**:
    - Python con bibliotecas como getpass, Tkinter o JSON.