**Universidad Nacional de Colombia - sede Bogotá**
**Facultad de Ingeniería**
**Departamento de Sistemas e Industrial**
**Curso: Ingeniería de Software 1 (2016701)**

**Consolidado de Requerimientos – Clasificación MoSCoW**

| Prioridad | Requisito | Estimación | Argumento |
|---|---|---|---|
| **MUST** | **RF_1.** El usuario podrá crear una cuenta e iniciar sesión con su correo institucional y una contraseña segura. | 5 | Es una funcionalidad esencial del sistema. Requiere autenticación, validación de formato y cifrado de contraseñas. |
| **MUST** | **RF_2.** El sistema deberá permitir la creación y gestión de perfiles de estudiante, garantizando una experiencia de uso personalizada y coherente. | 5 | Es clave para el control de usuarios y la personalización; implica formularios, validaciones y persistencia en base de datos. |
| **MUST** | **RF_3.** El sistema deberá almacenar la información en una base de datos segura, garantizando la confidencialidad e integridad de los datos. | 4 | Es fundamental para la seguridad del sistema; requiere diseño de base de datos, autenticación y respaldo. |
| **MUST** | **RF_4.** El administrador podrá gestionar recursos de apoyo (guías, videos, enlaces, documentos) para poner información útil y | 4 | Es una función central del administrador y requiere interfaz de carga, actualización y validación de archivos. |

| | | | |
|---|---|---|---|
| | actualizada a disposición de los estudiantes. | | |
| **MUST** | **RF_5.** El sistema deberá cifrar contraseñas y datos sensibles utilizando algoritmos criptográficos estándar (por ejemplo, SHA-256). | 3 | Es esencial para proteger los datos; requiere implementar métodos de cifrado y validación de seguridad. |
| **MUST** | **RNF_1.** El sistema deberá mantener disponibilidad del 95 % o superior durante el tiempo de operación. | 3 | Garantiza estabilidad y confiabilidad; requiere pruebas de carga y monitoreo de rendimiento. |
| **SHOULD** | **RF_6.** El sistema deberá permitir la recuperación de contraseña mediante un enlace enviado al correo electrónico registrado. | 3 | Es importante para la experiencia del usuario, pero puede implementarse después de las funciones básicas. |
| **SHOULD** | **RF_7.** El administrador podrá gestionar los perfiles de los estudiantes para mantener un control adecuado de los usuarios registrados. | 2 | Permite una administración eficiente, aunque no es crítica para la primera versión. |
| **SHOULD** | **RF_8.** El sistema deberá permitir la actualización y eliminación controlada de registros para mantener la información consistente y vigente. | 3 | Mejora la precisión de los datos y la integridad del sistema, pero puede implementarse luego del núcleo funcional. |

| | | | |
|---|---|---|---|
| **SHOULD** | **RNF_2.** El sistema deberá incluir pruebas de funcionamiento (unitarias e integradas) para garantizar un producto estable y confiable. | 3 | Mejora la calidad y mantenibilidad del software; puede desarrollarse en paralelo a las funciones críticas. |
| **COULD** | **RF_9.** El usuario podrá establecer notificaciones o recordatorios para cumplir tareas específicas de su rutina diaria. | 2 | Mejora la interacción y la constancia del usuario, pero no es esencial para la primera versión. |
| **COULD** | **RF_10.** Las gráficas de reportes semanales deberán ser claras y simples (barras o líneas), bien etiquetadas y con colores diferenciados por hábito. | 2 | Aporta visualización y análisis, pero puede añadirse tras completar las funciones base. |
| **COULD** | **RF_11.** Los formularios de registro de hábitos deberán usar controles intuitivos y fáciles de usar (deslizadores, botones gráficos, listas). | 2 | Mejora la usabilidad, pero no afecta la funcionalidad básica del sistema. |
| **COULD** | **RF_12.** La interfaz gráfica deberá incluir una barra de navegación con accesos a Inicio, Progreso, Recordatorios y Perfil. | 1 | Facilita la navegación del usuario, pero su ausencia no impide la operación del sistema. |
| **COULD** | **RF_13.** El usuario podrá consultar reportes | 3 | Permite al usuario analizar su progreso, aunque puede |

| | | | |
|---|---|---|---|
| | semanales para visualizar la evolución de sus hábitos. | | implementarse en una versión posterior. |
| **COULD** | **RF_14.** El sistema deberá funcionar bajo un modelo cliente-servidor con separación clara de lógica e interfaz. | 2 | Mejora la escalabilidad y mantenimiento, pero no es crítica en la versión inicial. |
| **WONT** | **RF_15.** Implementar autenticación con redes sociales (Google, Facebook). | — | No se incluirá en la versión actual por limitaciones de tiempo y alcance, pero puede considerarse en una versión futura. |
| **WONT** | **RF_16.** Incorporar gamificación (puntos, logros o niveles). | — | No es prioritaria para la primera versión; se puede evaluar en una futura actualización. |

Caso de uso:

| **CIFRAR CONTRASEÑAS Y DATOS SENSIBLES** | |
|---|---|
| **ACTORES**<br><br>Sistema biUNestar<br>Base de datos del sistema | **REQUERIMIENTO**<br><br>RF_5 – El sistema deberá cifrar contraseñas y datos sensibles utilizando algoritmos criptográficos estándar (por ejemplo, SHA-256). |
| **DESCRIPCIÓN**<br><br>Este caso de uso describe el proceso mediante el cual el sistema protege la información sensible de los usuarios, asegurando que contraseñas, correos institucionales y datos personales sean almacenados de forma cifrada e inaccesible para terceros. | |
| **PRECONDICIONES**<br><br>El usuario debe haber ingresado o actualizado información que incluya datos sensibles.<br>El sistema debe contar con el módulo de seguridad habilitado. | |

**FLUJO NORMAL**

1. El usuario ingresa una contraseña o dato sensible (por ejemplo, al registrarse o cambiar contraseña).
2. El sistema recibe el dato y lo somete al algoritmo de cifrado definido (SHA-256).
3. El sistema genera un hash cifrado del dato.
4. El sistema almacena únicamente el hash en la base de datos, sin conservar el valor original.
5. En futuras autenticaciones, el sistema compara el hash ingresado con el almacenado para validar el acceso.
6. Si coinciden, se concede el acceso o se confirma la operación.
7. El sistema registra la transacción en el log de seguridad.

**POSTCONDICIONES**

Los datos sensibles quedan protegidos mediante cifrado irreversible.
El sistema mantiene la integridad y confidencialidad de la información almacenada.

**NOTAS**

Se recomienda aplicar un “salting” único por usuario para fortalecer la seguridad.
El sistema debe cumplir con la política institucional de protección de datos personales.