



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

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

Requisito		Estimación	Argumento
RF_1	Registro y autenticación de usuarios	3	Nos costaría porque no dominamos la implementación de autenticación con tokens ni el manejo seguro de contraseñas; tendríamos que investigar librerías nuevas y configurar el back-end.
MUST			
RF_2	Creacion y gestion de perfiles personales	3	Como equipo, lo vemos normal ya que es básicamente un CRUD, pero igual debemos acordar validaciones y pruebas.
MUST			
RF_3	Evaluación de necesidades nutricionales	3	Es una tarea compleja porque requiere levantar infraestructura y hay que pedir ticket etc.
MUST			
RF_4	Definición de objetivos de salud	1	Es fácil para nosotros porque solo implica permitir al usuario escoger metas y guardarlas; no requiere lógica complicada.
MUST			
RF_5	Gestión de preferencias alimentarias	2	Tendríamos que diseñar bien cómo guardar las preferencias de los usuarios y hacer que se reflejen en el sistema de recomendaciones, lo que agrega cierta dificultad.
COULD			
RF_6	Manejo de restricciones dietéticas	3	Nos costaría definir todas las restricciones y validarlas contra los alimentos y recetas; puede haber muchos casos que no controlemos al inicio.
COULD			
RF_7	Recomendación basada en K-Nearest Neighbors	5	Esto sería de lo más difícil para el equipo porque la implementación de
MUST			

			algoritmos de recomendación como KNN, entrenarlos con una base de datos relacional es algo nuevo, tocaría profundizar un poco de teoría y práctica de machine learning.
RF_8	Generación de planes nutricionales personalizados	5	Generar planes completos sería complicado ya que deberíamos combinar recetas, balancear nutrientes y respetar restricciones; es algo que nunca hemos hecho como equipo.
MUST			
RF_9	Visualización de menús diarios	2	Puede no ser complejo porque es más de interfaz, aunque debemos aprender a mostrar los menús de forma clara.
MUST			
RF_10	Acceso a información Nutricional	2	El reto sería conectar la base de datos con la vista para que la información sea confiable
MUST			
RF_11	Seguimiento de progreso	3	Podría costarnos un poco porque implica manejar datos en el tiempo y generar gráficos, y no todos dominamos librerías de visualización.
COULD			
RF_12	Almacenamiento de historial	2	El equipo lo considera de dificultad media porque el manejo de base de datos puede llegar a ser tedioso y más en una con 3ra forma normal.
COULD			
RF_13	Comparativa de resultados	4	Implica registrar la información a lo largo del tiempo y constantemente solicitar la información bibliométrica del usuario
WONT			
RF_14	Guardado y recuperación de planes	3	Requiere programar la opción de guardar distintos planes y recuperarlos; como equipo lo vemos posible aunque con detalles que cuidar.
SHOULD			
RF_15	Exportación de datos	3	Nos costaría un poco generar archivos exportables en PDF

SHOULD			porque tendríamos que aprender a usar librerías específicas de exportación.
RF_16	Función de retroalimentación	2	La retroalimentación es sencilla, aunque debemos ponernos de acuerdo en qué datos recoger y cómo usarlos después. También podría no ser muy útil
WONT			
RFN_1	Seguridad de datos	4	Es difícil para nosotros porque implica conocer temas de seguridad, cifrado y buenas prácticas que no dominamos completamente.
MUST			
RFN_2	Usabilidad	3	Lo vemos de mediana dificultad porque debemos trabajar en diseño UX y accesibilidad, algo en lo que no todos tenemos experiencia.
SHOULD			
RFN_3	Rendimiento	3	Podría costarnos optimizar para que la aplicación no se vuelva lenta, ya que no tenemos mucha práctica con pruebas de rendimiento.
SHOULD			
RFN_4	Escalabilidad	4	Como equipo lo vemos complicado porque deberíamos aprender sobre escalabilidad y manejo de muchos usuarios concurrentes.
COULD			
RFN_5	Disponibilidad	3	El reto está en configurar el sistema para que tenga buena disponibilidad y no caiga, necesitaríamos investigar monitoreo y despliegues.
SHOULD			
RFN_6	Compatibilidad	2	Este requisito no es muy complejo, pero igual debemos probar en distintos navegadores y dispositivos, lo cual requiere tiempo.
SHOULD			
RFN_7	Mantenibilidad	2	Es relativamente fácil, aunque nos costaría mantener buenas prácticas de documentación y estilo de código si no lo planificamos
SHOULD			

			desde el inicio.
RFN_8	Tiempo de respuesta	3	Como equipo nos preocupa que el sistema responda rápido; tendríamos que aprender sobre caché y optimización de consultas.
SHOULD			
RFN_9	Precisión	5	Sería uno de los más difíciles porque implica validar la precisión con expertos y datasets reales, y eso supera nuestras capacidades actuales.
SHOULD			
RFN_10	Responsividad	2	Podemos manejarlo con frameworks responsive, pero igual debemos invertir tiempo en aprender CSS avanzado.
SHOULD			