



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

Ana Sofia Rodriguez Neira

Juliana Parra Caro

Carlos Javier Cuervo Baracaldo

Christian Alejandro Sanabria Pinzon

Entrega_03

CONSOLIDADO DE REQUISITOS- BUGBUSTERS

	Requisito	Estimación	Argumento
RF_1	El usuario podrá registrarse en el sistema proporcionando nombre, correo electrónico/teléfono y contraseña, para crear una cuenta personal.	3	Es complejo porque requiere implementar seguridad con hash de contraseñas y validación de datos sensibles
MUST			
RF_2	El usuario podrá iniciar sesión ingresando su correo electrónico y contraseña válidos, para acceder a su cuenta previamente registrada.	2	Es moderado, necesitamos sistema de autenticación con tokens y manejo de sesiones
MUST			
RF_3	El usuario podrá responder un cuestionario inicial con preguntas de selección múltiple y respuesta corta sobre hobbies, hábitos y metas, para que el sistema genere su perfil personalizado.	3	Es complejo porque requiere diseño de formulario dinámico y estructura de datos para perfiles
MUST			
RF_4	El sistema podrá generar un reto diario en formato texto corto basado en el perfil inicial del usuario, para ofrecer una actividad alineada con sus intereses.	4	Es muy complejo, necesitamos algoritmo de matching que no tenemos experiencia desarrollando
MUST			
RNF_1	El sistema podrá enviar un nuevo reto diario a cada usuario a las 8:00 a.m. hora local, en formato notificación push con texto.	4	Es complejo porque requiere levantar infraestructura de notificaciones push y sistema de scheduling
MUST			
RF_5	El sistema podrá asignar puntos al usuario por cada reto diario completado con evidencia válida, para motivar la participación continua.	2	Es sencillo, es lógica básica de suma y actualización de campos en base de datos
MUST			
RF_6	El sistema podrá calcular automáticamente la racha de días consecutivos cumplidos sumando los días con retos validados, para mostrar el avance del usuario.	3	Es moderado, requiere verificación de fechas consecutivas y manejo de timezones
MUST			
RF_7	El sistema podrá mostrar al usuario en pantalla principal su puntuación acumulada y racha actual, para mantener la motivación visible.	2	Es rutinario, componentes UI estándar que muestran consultas a la BD
MUST			

RF_8	El usuario podrá registrar la evidencia de un reto cumplido mediante tres formatos: subir una foto, escribir un texto o marcar un check automático, para confirmar la realización del reto.	4	Es complejo porque implica upload de archivos, procesamiento de imágenes y múltiples flujos
MUST			
RF_9	El sistema podrá validar que exista al menos un tipo de evidencia registrada antes de marcar un reto como "completado", para asegurar el cumplimiento.	2	Es sencillo, validación básica de campos requeridos en el backend
MUST			
RNF_2	El usuario podrá registrarse con la opción de registrarse con google para facilitar la creación de su cuenta personal.	3	Es moderado, requiere integración con OAuth2 de Google y manejo de flujos externos
MUST			
RF_10	El usuario podrá actualizar su perfil en cualquier momento modificando sus datos personales o respuestas del cuestionario, para mantener vigente la información de sus intereses.	2	Es rutinario, CRUD estándar para actualización de datos de usuario
MUST			
RF_11	El sistema podrá adaptar los retos diarios según las últimas actualizaciones de hobbies del usuario, para que los retos reflejen sus cambios recientes.	4	Es muy complejo, algoritmo dinámico que requiere machine learning que no dominamos
SHOULD			
RF_12	El sistema podrá generar un ranking semanal de los 5 usuarios con mayor puntuación y mostrarlo en un tablero público, para fomentar la competencia saludable.	3	Es moderado, consultas SQL complejas con agregaciones y paginación
SHOULD			
RF_13	El sistema podrá enviar una notificación push de advertencia con texto cuando falten 2 horas para el cierre del día y el usuario no haya completado su reto, para evitar la pérdida de progreso.	3	Es moderado, requiere jobs programados y lógica de notificaciones condicionales
SHOULD			
RF_14	El sistema podrá almacenar y mostrar un historial cronológico de actualizaciones de hobbies, para que el usuario pueda visualizar su progreso a lo largo del tiempo.	3	Es moderado, sistema de versionado de datos con historial de cambios
COULD			
RF_15	El sistema podrá mostrar al usuario un tablero personal con: número de retos completados, racha más larga, puntos acumulados y porcentaje de cumplimiento semanal, para que el usuario evalúe su progreso.	3	Es moderado, múltiples consultas a base de datos y diseño de dashboard
COULD			
RF_16	El sistema podrá mostrar al usuario una gráfica de barras con su evolución de rachas en el tiempo, para visualizar mejoras en su constancia.	3	Es moderado, integración con librerías de gráficos y preparación de datos históricos
COULD			
RF_17	El administrador podrá visualizar estadísticas generales de uso (número de usuarios activos, retos completados, rachas más largas), para monitorear la actividad del	4	Es complejo, dashboard administrativo con métricas en tiempo real y consultas CRUD para los datos
COULD			

	sistema.		
RF_18	El sistema podrá asignar retos predefinidos cuando el usuario no actualice sus hobbies durante una semana, para garantizar la continuidad de la experiencia.	2	Es sencillo, lógica condicional simple basada en fecha de última actualización
COULD			
RF_19	El sistema debe seleccionar a los 5 usuarios con mayor racha semanal como candidatos a proponer el reto global.	4	Muy complejo para MVP, requiere sistema de selección competitivo
WON'T			
RF_20	El sistema podrá elegir aleatoriamente a un único usuario entre los candidatos seleccionados, para definir quién propondrá el reto global de la semana.	3	Depende de RF_21, funcionalidad avanzada no esencial
WON'T			
RF_21	El sistema podrá validar que un usuario no sea elegido más de una vez en un mismo mes, para garantizar equidad entre participantes.	3	Lógica de equidad compleja, postergable
WON'T			
RF_22	El administrador podrá aprobar, editar o rechazar el reto global propuesto por un usuario, para asegurar que cumpla las normas de la comunidad.	4	Sistema de moderación manual, y fuera del scope inicial
WON'T			
RF_23	El sistema podrá enviar el reto global a todos los usuarios cada domingo a las 10:00 a.m., en formato de notificación push con texto y un botón de "Aceptar Reto".	3	Retos globales fuera del scope inicial
WON'T			
RF_24	El sistema podrá asignar 50 puntos adicionales al usuario por cada reto global completado, para reforzar la participación en eventos colectivos.	1	Depende de sistema de retos globales no implementado
WON'T			
RNF_3	El sistema podrá enviar una notificación push con vibración	2	Mejora UX no crítica para funcionalidad básica
WON'T			
RNF_4	El sistema podrá enviar una notificación push con el botón "Aceptar Reto", que abra la aplicación	3	Interacción compleja en notificaciones, baja prioridad
WON'T			
RF_27	El administrador podrá bloquear o eliminar retos globales inapropiados que contengan lenguaje ofensivo, violencia o contenido prohibido, para proteger a la comunidad.	4	Sistema de moderación automático muy complejo, y fuera del scope inicial
WON'T			