

## Evaluación Integral de la Calidad de los Requisitos

### Requerimientos Funcionales

#### RF-01

1. **Versión original:** El sistema deberá permitir al administrador gestionar las cuentas de usuario, diferenciando entre perfiles, comisión dictaminadora/RH y candidatos a docentes.
2. **Problema detectado:** El término "gestionar" es ambiguo (implica crear, leer, actualizar y borrar) y mezcla la gestión de cuentas con la asignación de roles en un mismo enunciado.
3. **Versión mejorada:** El sistema deberá permitir al administrador asignar un rol específico (Comisión Dictaminadora, Recursos Humanos o Candidato) durante el registro de una cuenta de usuario.
4. **Justificación técnica:** Se aísla la funcionalidad de asignación de privilegios para facilitar la implementación del control de acceso (RBAC).

#### RF-02

1. **Versión original:** El sistema deberá permitir al aspirante realizar la carga centralizada de sus documentos digitales en la plataforma.
2. **Problema detectado:** "Carga centralizada" es un término vago. No especifica qué tipo de documentos o el momento del proceso.
3. **Versión mejorada:** El sistema deberá permitir al usuario con rol de Aspirante cargar archivos digitales asociados a los requisitos de la vacante.
4. **Justificación técnica:** Define claramente el actor (Aspirante) y la acción concreta sobre los inputs del sistema.

#### RF-03

1. **Versión original:** El sistema deberá permitir al administrador visualizar los documentos cargados por los aspirantes para su cotejo visual.
2. **Problema detectado:** "Cotejo visual" es el propósito, no la función del sistema.
3. **Versión mejorada:** El sistema deberá permitir al administrador visualizar el contenido de los documentos digitales asociados al expediente de un aspirante.
4. **Justificación técnica:** Se centra en la capacidad del sistema de renderizar o presentar el archivo para su lectura, independientemente del proceso humano de cotejo.

#### RF-04

1. **Versión original:** El sistema deberá permitir al administrador registrar observaciones de aprobación o rechazo sobre documentos individuales.
2. **Problema detectado:** Mezcla la acción de comentar con la acción de cambiar un estado (aprobar/rechazar).
3. **Versión mejorada:** El sistema deberá permitir al administrador registrar comentarios de validación asociados a cada documento específico del expediente.

4. **Justificación técnica:** Separa la captura de datos (observaciones) de la lógica de negocio (cambio de estado), permitiendo trazabilidad.

#### RF-05

1. **Versión original:** El sistema será capaz de determinar si un candidato no entregó los documentos necesarios para su postulación.
2. **Problema detectado:** Redacción pasiva ("será capaz") y negativa ("no entregó"). Debe ser una validación activa.
3. **Versión mejorada:** El sistema deberá validar que el expediente contenga la totalidad de los documentos obligatorios antes de permitir el envío de la postulación.
4. **Justificación técnica:** Transforma una "capacidad" en una regla de validación bloqueante (gatekeeper) programable.

#### RF-06

1. **Versión original:** El sistema notificará a los administrativos en caso de haber una incongruencia en el expediente entregado.
2. **Problema detectado:** "Incongruencia" es subjetivo y difícil de automatizar sin definir reglas previas.
3. **Versión mejorada:** El sistema deberá enviar una alerta al administrador cuando un expediente sea marcado con estado "Rechazado" o "Pendiente de corrección".
4. **Justificación técnica:** Vincula la notificación a un evento de cambio de estado concreto en la base de datos, eliminando la ambigüedad.

#### RF-07

1. **Versión original:** El sistema podrá identificar candidatos dependiendo el puesto al que se estén postulando.
2. **Problema detectado:** "Podrá identificar" es ambiguo. Se refiere a filtrar o clasificar.
3. **Versión mejorada:** El sistema deberá permitir al administrador filtrar el listado de candidatos según la vacante seleccionada.
4. **Justificación técnica:** Define la función de búsqueda/filtrado necesaria para la interfaz de gestión.

#### RF-08

1. **Versión original:** El sistema deberá validar automáticamente que los archivos cargados cumplan con el formato solicitado y no se encuentren corruptos o vacíos.
2. **Problema detectado:** El original es aceptable, pero puede ser más preciso eliminando "automáticamente" (es implícito) y separando criterios.
3. **Versión mejorada:** El sistema deberá rechazar la carga de archivos que no cumplan con las extensiones y el tamaño máximo configurados.
4. **Justificación técnica:** Establece límites técnicos específicos (whitelist de extensiones, tamaño en bytes) verificables en el backend.

#### RF-09

1. **Versión original:** El sistema deberá restringir la vigencia y el acceso de las credenciales del aspirante estrictamente a la duración de la vacante o convocatoria activa.
2. **Problema detectado:** Mezcla autenticación con reglas de negocio.
3. **Versión mejorada:** El sistema deberá inhabilitar el inicio de sesión del usuario Aspirante una vez finalizada la fecha de vigencia de la convocatoria.
4. **Justificación técnica:** Describe el comportamiento exacto del módulo de autenticación frente a una condición temporal.

#### RF-10

1. **Versión original:** El sistema deberá permitir la desestimación de una candidatura únicamente cuando ésta no cumpla con los requisitos establecidos en el flujo de validación.
2. **Problema detectado:** Contiene una regla de negocio compleja ("únicamente cuando...").
3. **Versión mejorada:** El sistema deberá permitir al administrador cambiar el estatus de una candidatura a "Desestimada".
4. **Justificación técnica:** El requisito funcional es la capacidad de cambio de estado; las condiciones lógicas deben documentarse en las Reglas de Negocio, no en el RF.

#### RF-11

1. **Versión original:** El sistema deberá actualizar en tiempo real el estatus del trámite en el panel de control tras cada acción de validación o rechazo realizada por el administrador.
2. **Problema detectado:** "Tiempo real" es un requisito no funcional de rendimiento. "Panel de control" es diseño de interfaz.
3. **Versión mejorada:** El sistema deberá reflejar el nuevo estado del trámite inmediatamente después de confirmar la transacción de validación.
4. **Justificación técnica:** Asegura la consistencia de datos en la vista del usuario tras una operación de escritura.

#### RF-12

1. **Versión original:** El sistema deberá impedir la automatización de la decisión final de contratación, limitando sus funciones exclusivamente a la validación administrativa.
2. **Problema detectado:** Define lo que el sistema *no* debe hacer. Los requisitos deben ser positivos.
3. **Versión mejorada:** El sistema deberá requerir una confirmación manual por parte de un usuario con rol de Administrador para registrar el dictamen final de contratación.
4. **Justificación técnica:** Convierte una restricción negativa en una funcionalidad de aprobación explícita requerida para cerrar el proceso.

---

### Requisitos No Funcionales (RNF)

### RNF-01 (Seguridad)

1. **Versión original:** El sistema deberá gestionar el control de acceso basado en roles (RBAC) para asegurar que los aspirantes no tengan acceso al panel de administración.
2. **Problema detectado:** Redunda en la explicación del propósito ("para asegurar que...").
3. **Versión mejorada:** El sistema deberá restringir el acceso a las funciones administrativas exclusivamente a los usuarios con roles autorizados.
4. **Justificación técnica:** Establece la política de seguridad de "cero confianza" o privilegio mínimo.

### RNF-02 (Interoperabilidad)

1. **Versión original:** El sistema deberá operar de manera aislada sin requerir conexiones a APIs de terceros...
2. **Problema detectado:** Correcto, pero puede ser más directo sobre la autonomía de los datos.
3. **Versión mejorada:** El sistema deberá realizar todas las validaciones de datos utilizando exclusivamente su base de datos local, sin dependencias de servicios externos.
4. **Justificación técnica:** Define la arquitectura de datos como autocontenida (stand-alone).

### RNF-03 (Fiabilidad)

1. **Versión original:** El sistema deberá garantizar la integridad de los archivos recibidos mediante verificaciones automáticas que descarten archivos corruptos antes de su almacenamiento.
2. **Problema detectado:** Correcto. Se puede sintetizar.
3. **Versión mejorada:** El sistema deberá verificar la integridad de la estructura de bits de los archivos antes de procesar su almacenamiento.
4. **Justificación técnica:** Especifica el momento de la validación (pre-storage) para evitar "datos sucios".

### RNF-04 (Seguridad)

1. **Versión original:** El sistema deberá generar credenciales temporales para los aspirantes que expiren automáticamente al finalizar la convocatoria activa.
2. **Problema detectado:** Similar al RF-09, pero enfocado en la política de seguridad de la cuenta.
3. **Versión mejorada:** El sistema deberá configurar una fecha de expiración automática para las cuentas de usuario con rol de Aspirante.
4. **Justificación técnica:** Define un atributo de seguridad (Time-to-Live) asociado a la identidad del usuario.

### RNF-05 (Usabilidad)

1. **Versión original:** El sistema deberá proporcionar una interfaz de visualización que permita al administrador realizar el cotejo visual de los documentos sin necesidad de descargarlos.
2. **Problema detectado:** Bastante claro. Solo se ajusta para ser más formal.
3. **Versión mejorada:** El sistema deberá integrar un visor de documentos que permita la lectura de archivos en el navegador sin requerir descarga local.
4. **Justificación técnica:** Afecta la experiencia de usuario (UX) y reduce la fricción operativa al evitar descargas.

#### **RNF-06 (Infraestructura / Gestión de Datos)**

1. **Versión original:** El sistema deberá almacenar los documentos de los expedientes por medio de una object store.
2. **Problema detectado:** "Object store" es una decisión de diseño técnico (solución), no un requisito (necesidad). Falta la categoría del RNF.
3. **Versión mejorada:** El sistema deberá almacenar los archivos digitales en un repositorio escalable y desacoplado de la base de datos transaccional.
4. **Justificación técnica:** Permite al equipo de arquitectura decidir si usar S3, Blob Storage o un sistema de archivos, siempre que cumpla con la escalabilidad y desacoplamiento.