PRD: funcional

1. Visión y objetivo

Crear un **repositorio central e inteligente** para el Departamento de **Calidad** que permita:

- **Subir** certificados (PDF/fotos) de manera rápida (principalmente manual).
- **Clasificar y archivar** documentación por **Proyecto** y **PO (Pedido de compra)**.
- **Encontrar** documentos en segundos mediante búsqueda + filtros.
- Operar con un **estado** claro (pendiente/revisado/incidencia), incluyendo urgencia y antigüedad.
- **Notificar** automáticamente (in-app + email mock) al **owner** responsable.
- Soportar el flujo de **dossier** (descarga) como "feature de producto" aunque esté mockeado.

2. Alcance v1

2.1 Incluye (v1)

1. **Datos mock preexistentes (obligatorio)**

- El sistema debe arrancar con:
  - 3–5 **Proyectos** creados (mock).
  - Cada Proyecto con 3–8 **POs** creadas (mock).
  - (recomendado) 10–30 **certificados** ya cargados para demo.
- **Regla de modelo (obligatoria):**
  - **1 Proyecto : N POs**
  - **1 PO : N Certificados**

1. **Ingesta manual de certificados (obligatorio)**

- Subida de PDF/foto desde web (`Subir certificado`).
- Tras subir, **mock de extracción**:

    - loader “leyendo documento…”
    - autocompletar **Proveedor** (string) si “se detecta”
- Campos obligatorios (no se puede guardar si faltan):
    - Proyecto (dropdown)
    - PO (dropdown ligado al proyecto)
    - Proveedor (auto por mock, pero editable)
    - Fecha de recepción del documento
- Campos opcionales:
    - Nº albarán
    - Tipo de documento (Certificado / Albarán / Factura / Otro)
    - Observaciones
- UX de errores: validación inline + mensaje claro (“Falta PO”, “Selecciona proyecto”, etc.).

1. **Clasificación y archivado**

- Todo documento queda asociado a:
    - **Proyecto**
    - **PO**
    - (y opcionalmente) proveedor/albarán/tipo
- Estructura de navegación:
    - Proyecto → POs → Certificados

1. **Búsqueda y filtros**

- Búsqueda “clásica” por:
    - Proyecto, PO, Proveedor, Albarán, texto libre
- Filtros:
    - Estado (pendiente/revisado/incidencia)
    - Urgencia (normal/urgente)
    - Rango fechas
    - Antigüedad (pendiente desde X días)

- Resultados con vista útil:
  - lista + preview + panel lateral de metadatos.

1. **Búsqueda “AI-assisted” que propone filtros (recomendado, mock aceptado)**

- Input en lenguaje natural:
  - ej. “certificados pendientes del proyecto P-102, proveedor Tubacex, urgentes”
- La “IA” genera chips de filtros (propuestos) y los aplica.
- Implementación permitida:
  - parser simple / reglas / hardcode por demo (lo importante es el UX).

1. **Estado operativo + extras obligatorios**

   Estados por certificado:

- Pendiente de clasificar (solo si lo necesitáis; ver nota abajo)
- Pendiente de revisión
- Revisado/validado
- Incidencia

Extras obligatorios:

- **Urgencia**: Normal / Urgente
- **Aging**: “pendiente desde X días”

> Nota: Como habéis pedido campos obligatorios, “Pendiente de clasificar” puede omitirse si no aporta. Si se incluye, debe tener un caso claro (p.ej. documento subido sin PO — pero eso solo si decidís permitirlo; por defecto NO).

1. **Notificaciones (in-app + email mock)**

- Notificación automática cuando:
  - se sube un nuevo documento a una PO
  - se crea una incidencia
- Requisitos:
  - cada Proyecto (y opcionalmente cada PO) tiene un **Owner**

    - notificación al owner:
        - in-app (bandeja)
        - email **mock** con plantilla auto-generada
    - dropdown para seleccionar destinatarios manualmente (multi-select):
        - “Enviar también a…” + previsualización del email.

1. **Dossier / descarga (mock permitido)**

- Botón `Descargar dossier` en:
    - Proyecto
    - PO
- Comportamiento aceptado para v1:
    - botón → modal → toast “generado/descargado”
- (opcional) selector “solo revisados / todo / solo incidencias”.

1. **0 resultados (obligatorio)**

   Cuando una búsqueda no devuelve documentos:

- Panel con 3 acciones separadas:
    1. `Subir certificado ahora`
    2. `Crear incidencia`
    3. `Notificar internamente` (email mock con plantilla)
- Cada acción debe dejar rastro (log/timeline visual).

2.2 Opcional (si da tiempo; mock aceptado)

1. **Foto móvil / escaneo**

- Flujo tipo “escaneo” + “detección de campos” + confirmación.
- Puede ser 100% mock (UX convincente).

1. **Botón “Descarga real”**

- Un botón mock adicional que “simula” generar ZIP/PDF.

2.3 Fuera de alcance v1

- Ingesta por email real (se elimina).
- Integración con Navision/Dynamics.
- Generación automática de colada corta / lectura de albaranes / trazabilidad total.
- OCR/IDP real con precisión alta (solo mock/heurísticas).
- Automatización real de seguimiento de incidencias con sistemas externos.

---

3. Usuarios y roles

3.1 Internos

1. **Calidad (usuario principal)**

- Sube certificados
- Busca y filtra
- Revisa/valida documentos
- Crea incidencias
- Descarga dossier (mock)

1. **Owner / Responsable (por proyecto)**

- Recibe notificaciones
- Decide acciones sobre incidencias
- Puede ser Mónica / Porfirio / Jefe proyecto (según organización)

1. **Admin (opcional en demo)**

- Configura owners
- Crea proyectos/POs mock (o solo se carga por JSON)

---

4. Userflow completo TO-BE (visión funcional)

4.1 Setup inicial (mock)

1. El sistema muestra una lista de **Proyectos** pre-creados.
2. Al entrar a un Proyecto se ven sus **POs**.

3. Cada Proyecto muestra su **Owner** (editable, si hay pantalla de configuración).

4.2 Subida de certificado (manual)

1. Usuario entra en `Calidad > Subir certificado`.
2. Selecciona **Proyecto** (dropdown).
3. Selecciona **PO** del proyecto (dropdown).
4. Sube PDF/foto.
5. El sistema muestra:
   - “Leyendo documento…” (mock)
   - Autorellena **Proveedor** si “detectado”
6. Usuario completa obligatorios y guarda.
7. Resultado:
   - Certificado aparece en la PO
   - Se crea evento en timeline
   - Se dispara notificación al owner (in-app + email mock)

4.3 Navegación por repositorio

1. Usuario entra en un Proyecto.
2. Abre una PO.
3. Ve lista de certificados:
   - estado, urgencia, fecha, aging
4. Abre un certificado:
   - preview + metadatos
   - acciones: cambiar estado / marcar incidencia / añadir comentario

4.4 Búsqueda y filtros (clásico)

1. Usuario usa buscador en cabecera (por proveedor, albarán, etc.).
2. Aplica filtros por estado/urgencia/fecha/aging.
3. Accede rápidamente al documento correcto.

4.5 Búsqueda “IA” que propone filtros (mock)

1. Usuario escribe “Necesito certificados pendientes del proveedor X del proyecto Y”.
2. El sistema muestra chips:
   - Proyecto=Y, Proveedor=X, Estado=Pendiente
3. Usuario confirma/edita chips y ve resultados.

4.6 0 resultados

1. Usuario busca y obtiene 0 resultados.
2. El sistema sugiere:
   - subir documento
   - crear incidencia
   - notificar internamente (email mock)
3. Log visual de lo que hizo (evento).

4.7 Dossier (mock)

1. Usuario entra en Proyecto o PO.
2. Click `Descargar dossier`.
3. Modal (elige alcance: todo / revisados).
4. Confirmación + toast.

---

5. Requisitos funcionales (v1)

5.1 Módulo Proyectos y POs (mock)

- Lista de proyectos con:
  - nombre, owner, nº POs, nº certificados, nº pendientes/incidencias
- Dentro de proyecto:
  - lista de POs con:
    - proveedor principal (opcional)
    - contadores (docs, pendientes, incidencias)

        - owner (si se soporta por PO)
- CRUD:
    - para demo, puede ser “solo lectura” + edición de owner
    - crear proyecto/PO desde UI es opcional (o hardcode JSON)

5.2 Módulo Documentos (certificados)

- Subida de archivo (PDF/jpg/png)
- Metadatos obligatorios + opcionales
- **Mock extracción** al subir:
    - autocompletar proveedor + “confidence badge” (opcional)
- Vista detalle de documento:
    - preview
    - metadatos editables
    - estado/urgencia
    - historial de eventos

5.3 Estados, urgencia y aging

- Estado editable desde:
    - lista
    - detalle
- Urgencia editable (chip o dropdown)
- Aging calculado:
    - basado en fecha recepción doc
    - mostrado en listas (“Pendiente desde 6 días”)
- Reglas UI:
    - si `Incidencia` → requiere comentario (mínimo 1 línea)

5.4 Búsqueda y filtros

- Búsqueda global (input) con:
    - matching por proveedor, PO, albarán, texto libre

- Filtros:
  - estado, urgencia, fecha, aging
- UX:
  - chips removibles
  - “guardar búsqueda” opcional (mock)

5.5 Búsqueda IA (mock)

- Input NLQ + botón `Aplicar`
- Resultado:
  - chips propuestos + explicación breve (“He aplicado: Proyecto=P-102, Proveedor=Tubacex...”)
- Permitir al usuario editar chips manualmente.

5.6 Notificaciones + email mock

- Bandeja in-app:
  - “Nuevo documento en PO-XXXX”
  - “Incidencia creada en PO-XXXX”
- Email mock:
  - plantilla auto-generada (subject + body)
  - preview antes de enviar
  - dropdown destinatarios (multi-select) + owner por defecto
- Log:
  - “Email enviado (mock) a X, Y”

5.7 Incidencias

- Crear incidencia desde:
  - documento
  - vista 0 resultados
- Campos mínimos:
  - motivo (dropdown: falta documento / documento incorrecto / otro)

    - comentario
    - urgencia
- Estado de incidencia:
    - Abierta → En curso → Cerrada (mock)
- Acciones:
    - “Notificar internamente” (email mock)
    - “Marcar resuelta” (mock)

5.8 Dossier / descarga (mock)

- Botón visible y convincente
- Modal con alcance:
    - todo / revisados / pendientes
- Confirmación + toast + evento en timeline.

5.9 Auditoría visual (muy recomendado)

- Timeline por Proyecto/PO/documento:
    - subida, cambio de estado, incidencia, notificación
- Puede ser completamente front (array de eventos).

---

6. Requisitos no funcionales

- **Calidad de demo / UX**
    - UI limpia, “industrial”, con buen gusto.
    - Empty states muy cuidados (incl. 0 resultados).
    - Microcopy claro (evitar jerga técnica innecesaria).
- **Priorización**
    - Mejor 6 features excelentes que 12 a medias.
- **Performance**
    - Respuesta inmediata (aunque todo sea mock).
- **Trazabilidad**

    - Todo click relevante deja un evento visible (demo-friendly).
- **Mock vs Real (obligatorio documentarlo)**
    - Sección al final del PRD y/o badges en UI indicando:
        - **Mock** (extracción proveedor, IA filtros, emails, descarga)
        - **Real** (navegación, validaciones, estados, UI)

---

7. “Mock vs Real” (para dejarlo cristalino)

Real (debe funcionar):

- navegación Proyecto → PO → Certificados
- subida manual + validación campos obligatorios
- estados/urgencia/aging
- búsqueda clásica + filtros
- 0 resultados con acciones (aunque lo que pase después sea mock)
- bandeja in-app (aunque sea mock data)

Mock permitido (ideal):

- extracción proveedor al subir
- búsqueda IA que propone filtros
- email envío (preview + log)
- descarga dossier (botón + confirmación)

Opcional:

- escaneo móvil (mock)