



**Luis Imaicela**  
PROYECTO FINAL DAM  
Fase Análisis  
11/11/2025

**Santiago Roman Viguera**  
TUTOR DEL PROYECTO



# LUIS IMAICELA

Diseño y Comunicación

## INDICE

CRONOGRAMA DE ACTIVIDADES .....	1
FASE DE ANÁLISIS	
1. ANÁLISIS DE REQUERIMIENTOS FUNCIONALES	
1.1 Autenticación y gestión de usuarios .....	2
1.2. Gestión de trabajos (core de la app).....	3
1.3. Flujo de estados y seguimiento.....	4
1.4. Comentarios, revisiones y comunicación interna .....	4
1.5. Consulta y filtrado de trabajos.....	4
1.6. Historial y auditoría.....	5
2. REQUERIMIENTOS NO FUNCIONALES	
2.1. Usabilidad y experiencia de usuario .....	6
2.2. Rendimiento .....	6
2.3. Seguridad .....	6
2.4. Confiabilidad y mantenimiento .....	7
2.5. Portabilidad y despliegue .....	7
3. IDENTIFICACIÓN DE ROLES	
3.1. Actor: Administrador .....	8
3.2. Actor: Diseñador .....	8
4. ELABORACIÓN DE CASOS DE USO UML	
4.1. Actores del sistema .....	9
4.2. Lista de casos de uso principales .....	9
4.3. Ejemplo de especificaciones textuales de casos de uso .....	9
4.4. Representación diagrama UML .....	11

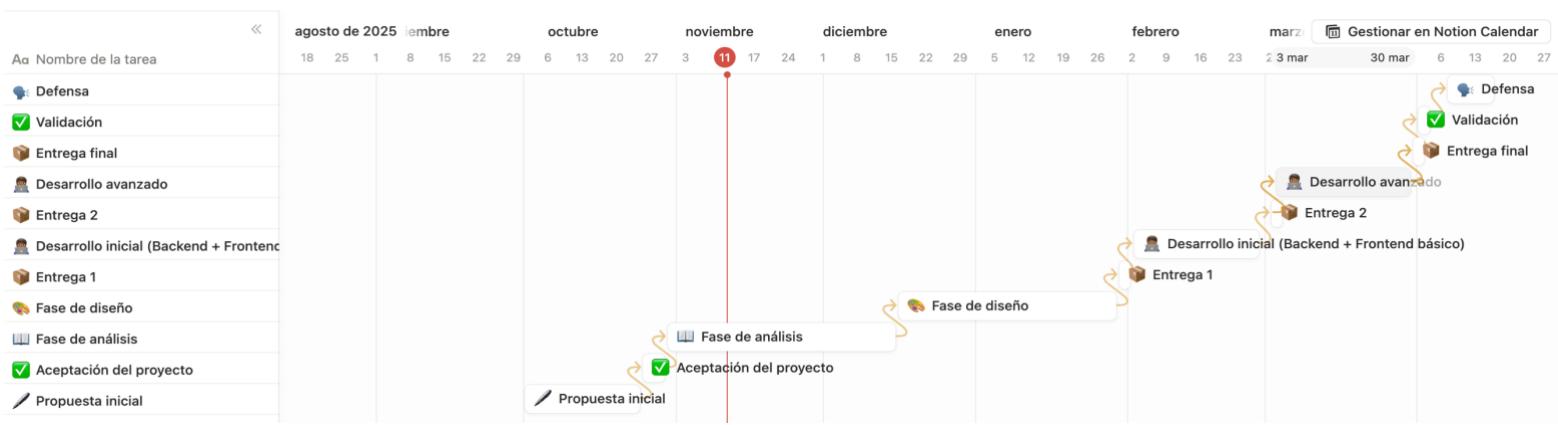


# LUIS IMAICELA

Diseño y Comunicación

## Proyecto final DAM

### CRONOGRAMA DE ACTIVIDADES



Nombre de la tarea	Plazo
Defensa	7 de abril de 2026 → 16 de abril de 2026
Validación	1 de abril de 2026 → 3 de abril de 2026
Entrega final	31 de marzo de 2026
Desarrollo avanzado	3 de marzo de 2026 → 30 de marzo de 2026
Entrega 2	2 de marzo de 2026
Desarrollo inicial (Backend + Frontend básico)	2 de febrero de 2026 → 27 de febrero de 2026
Entrega 1	30 de enero de 2026
Fase de diseño	16 de diciembre de 2025 → 29 de enero de 2026
Fase de análisis	30 de octubre de 2025 → 15 de diciembre de 2025
Aceptación del proyecto	25 de octubre de 2025 → 29 de octubre de 2025
Propuesta inicial	1 de octubre de 2025 → 24 de octubre de 2025



## FASE DE ANÁLISIS

### 1. ANÁLISIS DE REQUERIMIENTOS FUNCIONALES

---

La app “**DesignWorks**” permite gestionar los trabajos y encargos de un estudio de diseño gráfico. Su objetivo es centralizar la organización de proyectos y la comunicación entre los miembros del equipo. Los requisitos funcionales se dividen por módulos.

#### 1.1. Autenticación y gestión de usuarios

##### RF1. Registro / alta de usuarios (solo backend / admin).

- Por razones de tiempo y simplicidad, la creación de usuarios se realizará exclusivamente desde el **backend** (fuera de la app móvil).
- El sistema debe permitir dar de alta usuarios con: nombre, email, rol (Administrador/Diseñador), contraseña.
- La contraseña se almacenará en la BD cifrada (hash).
- Solo un usuario con rol **Administrador** (en el backend) podrá **crear, modificar o desactivar** usuarios.
- En la aplicación móvil, los usuarios únicamente podrán **iniciar sesión** con las credenciales que se les haya asignado previamente.

##### RF2. Inicio de sesión con JWT

- La app debe ofrecer una pantalla de **login** en la que el usuario introduce email y contraseña.
- El backend valida las credenciales y, si son correctas, devuelve un **token JWT**.
- La app almacena el token de forma segura (`flutter_secure_storage`) y lo enviará en cada petición.
- Si el token es inválido o caducado, la app mostrará un mensaje y devolverá al login.

##### RF3. Gestión básica de perfil

- El usuario podrá consultar sus datos básicos (nombre, email, rol) desde la app.
- El usuario podrá cambiar su contraseña (opcional como mejora futura pero me ha parecido interesante mencionarlo).



## 1.2. Gestión de trabajos (core de la app)

### RF4. Creación de trabajo (Administrador)

- El Administrador puede crear un nuevo trabajo indicando como mínimo:
  - Título del trabajo.
  - Cliente.
  - Descripción breve / briefing.
  - Prioridad (baja, media, alta, urgente).
  - Fecha de creación (automática).
  - Fecha límite (opcional).
- Al crear el trabajo, su **estado inicial** será “Creado”.

### RF5. Edición y eliminación de trabajos (Administrador)

- El Administrador puede editar los datos de un trabajo mientras no esté en estado “Entregado” o “Cancelado”.
- El Administrador puede marcar un trabajo como “Cancelado” si finalmente no se realiza (en lugar de borrarlo físicamente).

### RF6. Asignación de participantes a un trabajo

- El Administrador podrá seleccionar uno o varios diseñadores y asignarlos a un trabajo.
- Se registrará el rol del diseñador en ese trabajo (por ejemplo, diseñador principal, apoyo, ilustrador, maquetador...).
- Los diseñadores solo verán en “Mis trabajos” aquellos en los que sean participantes

### RF7. Definición de requisitos / briefs del trabajo

- Para cada trabajo, se podrán definir uno o varios **requisitos** (briefs parciales o puntos clave):
  - Descripción del requisito.
  - Posible adjunto (URL a referencia, archivo, etc.).
- Los requisitos serán visibles para todos los participantes asignados.



### 1.3. Flujo de estados y seguimiento

#### RF8. Gestión de estados del trabajo

Estados principales:

- Creado
- En progreso
- En revisión
- Entregado
- Cancelado

Reglas mínimas del flujo:

- De “Creado” → se puede pasar a “En progreso” o “Cancelado”.
- De “En progreso” → se puede pasar a “En revisión” o “Cancelado”.
- De “En revisión” → se puede pasar a “En progreso” (si hay cambios) o “Entregado”.
- “Entregado” y “Cancelado” se consideran estados finales (no se puede cambiar más, esto lo he establecido así por ahora, en un futuro puede cambiarse).

#### RF9. Cambio de estado (Administrador y Diseñador)

- Un **Diseñador participante** puede cambiar el estado del trabajo si tiene permisos (por ejemplo, de “En progreso” a “En revisión”).
- El **Administrador** puede forzar cambios de estado en cualquier momento.
- Cada cambio de estado debe registrar en la tabla de historial:
  - Trabajo.
  - Estado nuevo.
  - Fecha y hora.
  - Usuario que realiza el cambio.
  - Motivo o comentario de cambio (opcional pero muy útil en un estudio real).

### 1.4. Comentarios, revisiones y comunicación interna

#### RF10. Comentarios / revisiones sobre el trabajo

- Los participantes (Administrador y Diseñadores asignados) pueden añadir comentarios al trabajo.
- Cada comentario tiene: texto, usuario, fecha/hora.
- Los comentarios se listan cronológicamente, a modo de “timeline” del trabajo.



## 1.5. Consulta y filtrado de trabajos

### RF11. Vista general de trabajos (Administrador)

- Lista de todos los trabajos registrados.
- Posibilidad de filtrar por: estado, prioridad, cliente, rango de fechas.
- Posibilidad de ordenar por fecha de creación, fecha límite, prioridad, etc.

### RF12. “Mis trabajos” (Diseñador)

- Cada Diseñador verá una lista de los trabajos en los que está asignado.
- Podrá filtrar por estado (por ejemplo, ver solo “En progreso”).

### RF13. Detalle de trabajo

- Al entrar al detalle se verá:
  - Datos básicos: título, cliente, prioridad, descripción.
  - Requisitos / briefs.
  - Participantes.
  - Estado actual.
  - Historial de estados.
  - Comentarios.

## 1.6. Historial y auditoría

### RF14. Consulta de historial de estados

- Desde el detalle del trabajo, se podrá ver el **historial de estados** con fecha y usuario.
- También habrá un endpoint tipo “GET /historial/{trabajoid}” para recuperar esta información.



## 2. REQUERIMIENTOS NO FUNCIONALES

---

### 2.1. Usabilidad y experiencia de usuario

- **RNF1. Interfaz clara y consistente:** diseño basado en **Material Design 3** con modo claro/oscuro, pensada para uso intensivo en un estudio.
- **RNF2. Navegación simple:**
  - Menú principal: “Mis trabajos”, “Todos los trabajos” (solo admin), “Perfil”.
  - Flujo directo: lista → detalle → cambio de estado / comentario.
- **RNF3. Validación de formularios:** campos obligatorios señalados, mensajes de error claros (ej. “Título obligatorio”, “Formato de email inválido”).

### 2.2. Rendimiento

- **RNF4. Tiempo de respuesta:** las operaciones comunes (consultar trabajos, cambiar estado, añadir comentario) deberían responder en **< 300 ms** en condiciones normales de red local o WiFi estándar.
- **RNF5. Eficiencia en consultas:** uso de índices adecuados en la BD (por ejemplo en campos de búsqueda frecuentes: estado, prioridad, fecha, cliente).

### 2.3. Seguridad

- **RNF6. Autenticación y autorización:**
  - Acceso a la API mediante **JWT**.
  - Protección de endpoints según rol (Admin vs Diseñador) mediante Spring Security.
- **RNF7. Protección de credenciales:**
  - Contraseñas cifradas con algoritmo robusto (BCrypt).
  - Token almacenado en “flutter\_secure\_storage” en el móvil.
- **RNF8. Control de permisos:** un usuario solo puede ver y modificar lo que le corresponde (por ejemplo, un diseñador no ve trabajos de otros si no está asignado).



## 2.4. Confiabilidad y mantenimiento

- **RNF9. Integridad de datos:**
  - Uso de transacciones en operaciones críticas (creación de trabajo + asignación de participantes, cambios de estado...).
- **RNF10. Registro de errores (logging):**
  - La API debe registrar errores y excepciones para facilitar el diagnóstico.
- **RNF11. Mantenibilidad:**
  - Arquitectura por capas (controlador, servicio, repositorio).
  - Código documentado y versionado en GitHub.

## 2.5. Portabilidad y despliegue

- **RNF12. Portabilidad de backend:**
  - Backend empaquetado con **Docker Compose** (contenedor API + BD), lo que permite ejecutarlo en diferentes máquinas de forma homogénea.
- **RNF13. Portabilidad de frontend:**
  - App desarrollada en **Flutter** preparada para Android, con posibilidad futura de iOS sin reescribir toda la lógica.



## 3. IDENTIFICACIÓN DE ROLES

---

### 3.1. Actor: Administrador

#### Descripción

Persona responsable de la organización de los trabajos dentro del estudio de diseño (por ejemplo, el jefe de estudio o project manager).

#### Objetivos

- Tener una visión global de todos los trabajos.
- Asignar tareas a los diseñadores.
- Supervisar el avance y garantizar que se cumplan los plazos.

#### Permisos principales

- Crear, editar y cancelar trabajos.
- Asignar y modificar participantes.
- Definir o editar requisitos del trabajo.
- Cambiar estados de cualquier trabajo.
- Consultar todos los historiales y comentarios.
- Dar de alta o desactivar usuarios (opcional según alcance final).

### 3.2. Actor: Diseñador

#### Descripción

Miembro del equipo creativo del estudio que realiza los trabajos asignados.

#### Objetivos

- Ver claramente qué trabajos tiene asignados, con prioridades y plazos.
- Comunicar avances y dudas mediante comentarios.
- Indicar cambios de estado acorde al flujo real de trabajo.

#### Permisos principales

- Ver la lista de “Mis trabajos” y el detalle de cada uno.
- Ver requisitos, participantes y comentarios de sus trabajos.
- Añadir comentarios / revisiones.
- Cambiar estado de sus trabajos dentro del flujo permitido (por ejemplo, de “En progreso” a “En revisión”).
- Consultar el historial de estados de sus trabajos.

*(Importante: el diseñador NO puede crear trabajos nuevos)*



## 4. ELABORACIÓN DE CASOS DE USO UML

---

### 4.1. Actores del sistema

- **Administrador**
- **Diseñador**
- (Opcional) **Sistema de Autenticación** como actor externo que valida credenciales.

### 4.2. Lista de casos de uso principales

1. **CU1 – Iniciar sesión** (Administrador / Diseñador)
2. **CU2 – Gestionar usuarios** (Administrador) – opcional, según alcance.
3. **CU3 – Crear trabajo** (Administrador)
4. **CU4 – Editar / cancelar trabajo** (Administrador)
5. **CU5 – Asignar participantes a un trabajo** (Administrador)
6. **CU6 – Definir requisitos del trabajo** (Administrador)
7. **CU7 – Consultar todos los trabajos** (Administrador)
8. **CU8 – Consultar mis trabajos** (Diseñador)
9. **CU9 – Ver detalle de trabajo** (Administrador / Diseñador)
10. **CU10 – Cambiar estado de trabajo** (Administrador / Diseñador)
11. **CU11 – Añadir comentario / revisión** (Administrador / Diseñador)
12. **CU12 – Consultar historial de estados** (Administrador / Diseñador)

### 4.3. Ejemplo de especificaciones textuales de casos de uso

#### **CU3 – Crear trabajo**

- **Actor principal:** Administrador
- **Objetivo:** Registrar un nuevo trabajo en el sistema para su posterior asignación y seguimiento.
- **Precondiciones:**
  - El Administrador ha iniciado sesión.
- **Flujo principal:**
  1. El Administrador accede a la opción “Nuevo trabajo”.
  2. El sistema muestra un formulario de creación.
  3. El Administrador introduce título, cliente, descripción, prioridad y fecha límite (opcional).
  4. El Administrador confirma la creación.
  5. El sistema valida los datos (campos obligatorios, formato de fecha).
  6. El sistema guarda el trabajo en la BD con estado inicial “Creado” y fecha de creación actual.
  7. El sistema muestra un mensaje de confirmación y el nuevo trabajo aparece en la lista.



- **Postcondiciones:**
  - El trabajo queda registrado y disponible para su asignación y gestión posterior.
- **Flujos alternativos:**
  - Si hay errores de validación, el sistema informa de los campos incorrectos y vuelve al formulario sin perder la información introducida.

## CU8 – Consultar mis trabajos

- **Actor principal:** Diseñador
- **Precondición:**
  - El Diseñador ha iniciado sesión correctamente.
- **Flujo principal:**
  1. El Diseñador selecciona la opción “Mis trabajos”.
  2. El sistema consulta en la BD todos los trabajos en los que el usuario está asignado.
  3. El sistema muestra la lista filtrable por estado y prioridad.
- **Postcondición:**
  - El Diseñador visualiza únicamente los trabajos en los que participa.

## CU10 – Cambiar estado de trabajo

- **Actor principal:** Diseñador (o Administrador)
- **Precondiciones:**
  - El usuario ha iniciado sesión.
  - El usuario es participante del trabajo (o es Administrador).
- **Flujo principal:**
  1. El usuario abre el detalle de un trabajo.
  2. El usuario selecciona “Cambiar estado”.
  3. El sistema muestra los estados a los que se puede cambiar desde el estado actual.
  4. El usuario elige el nuevo estado e introduce opcionalmente un motivo/comentario.
  5. El sistema valida que la transición es permitida según las reglas de negocio.
  6. El sistema actualiza el estado del trabajo y registra la entrada en “historial\_estados”.
  7. El sistema muestra confirmación.
- **Flujos alternativos:**
  - 5a. Si la transición no está permitida, el sistema muestra un mensaje de error y no realiza cambios.



## LUIS IMAICELA

Diseño y Comunicación

### 4.4. Representación diagrama UML

