| **Proyecto** | : | DocuFlow |
| --- | --- | --- |
| **Ciudad** | : | Santiago |
| **Docente de Práctica** | : | Cristian Espinoza |
| **Alumnos** | : | Camila Orozco  Darío Coñuenao  Gary Hernández |
| **Fecha** | : | 17.Septiembre.2025 |



Tabla de contenido

[Proyecto DocuFlow 4](#_heading=h.b5yzrqi49wkf)

[Contexto 4](#_heading=h.ik4ilyrrxtus)

[Alcance General 4](#_heading=h.5fah9pvo8u7b)

[Enfoque en Duoc UC 5](#_heading=h.3zyt2mao4hxc)

[Escalabilidad Posterior 5](#_heading=h.nl2m9wqmmepv)

[Solución 6](#_heading=h.wd1bsnd8x1zw)

[Propuestas y Alcances 6](#_heading=h.9x8k2ccilrb5)

[Funcionalidades Clave del Modelo 6](#_heading=h.8esbvvi5r146)

[Diagramas 8](#_heading=h.6m3lmca0prfw)

[Diagrama de BD 8](#_heading=h.83k3qmd71yol)

[2. Región 8](#_heading=h.yuz7snn2i1t0)

[3. Comuna 8](#_heading=h.c5ab0x34ufh1)

[4. Empresa 9](#_heading=h.uskwpdxksa9o)

[5. Proyecto 9](#_heading=h.sjdt28rit4qh)

[6. Rol 9](#_heading=h.aqegnnfcrtls)

[7. Usuario 9](#_heading=h.bryhvrgq2byx)

[8. Perfil 9](#_heading=h.uh5baeci5udu)

[9. Usuario\_Proyecto 9](#_heading=h.zfvynsogmr1b)

[10. Fase 9](#_heading=h.91ws85cwq7m)

[11. Documento 9](#_heading=h.tktoymb74i07)

[12. Fase\_Documento 9](#_heading=h.e2nxajmqbkes)

[13. Privilegio 10](#_heading=h.xqnx4mrofk00)

[Diagrama de Arquitectura 11](#_heading=h.q1nukf4w3h0b)

[Propuesta de Valor y Factor Diferenciador 13](#_heading=h.vnw56gt92w9q)

[Arquitectura y Tecnologías del Proyecto 13](#_heading=h.efqafe6s6il7)

[Arquitectura de tres capas 14](#_heading=h.zdrf0ps8xluq)

[Capas de la Arquitectura 14](#_heading=h.pywk5n4yobhj)

[Despliegue en la Nube 15](#_heading=h.pcdijt9ywj3j)

[Cronograma DocuFlow 16](#_heading=h.lm1dcx5h3fxb)

# Proyecto DocuFlow

## Contexto

En el desarrollo de proyectos de software, una de las principales falencias detectadas es la ausencia de una documentación técnica clara, estandarizada y actualizada. Esta situación genera dificultades en la continuidad de los proyectos, en la incorporación de nuevos integrantes a los equipos de trabajo y en la mantención futura de los sistemas.

En particular, las pequeñas y medianas empresas tecnológicas suelen carecer de metodologías estructuradas que aseguren la trazabilidad y la calidad de la información generada. Esto repercute directamente en la eficiencia de los equipos, la comunicación interna y la satisfacción de los usuarios.

El proyecto DocuFlow se plantea dentro de este contexto simulado, replicando condiciones reales de empresas que necesitan optimizar sus procesos de documentación técnica y gestión de proyectos, con el fin de garantizar orden, claridad y continuidad en los sistemas desarrollados.

## Alcance General

DocuFlow es un modelo de gestión y estandarización de documentación técnica, diseñado para ser utilizado por cualquier empresa, sin importar su rubro o tamaño. Para lograr esta versatilidad, el modelo incluye módulos administrativos (o mantenedores) que permiten a un gestor personalizar el sistema. A través de estos módulos, es posible adaptar las plantillas de documentos, configurar los permisos de los usuarios, estructurar los proyectos y definir los procesos de validación y aprobación según las necesidades específicas de cada organización.

Si bien el diseño de la solución apunta a ser un modelo replicable y escalable para su implementación en cualquier tipo de organización, el alcance de este proyecto se centrará en la implementación y validación de un prototipo funcional dentro del marco académico de Duoc UC. Esto asegura que los objetivos sean claros y alcanzables, y que la demostración de sus capacidades se realice en un entorno controlado y pertinente para el contexto del proyecto.

Las acciones concretas que definen el alcance son:

* Se desarrollarán plantillas estandarizadas y se generará documentación real, como avances, cronogramas y la documentación técnica del proyecto (ej. requerimientos y especificaciones).
* Se implementará un sistema de gestión de permisos para los distintos roles de usuario y se establecerá un flujo de trabajo para la carga, revisión y aprobación de la documentación.
* Se utilizará un proyecto de validación para demostrar la aplicabilidad, consistencia y adaptabilidad del modelo.
* El prototipo integrará funcionalidades reales para el registro del historial de carga de los documentos y la visualización del avance del proyecto.

## Enfoque en Duoc UC

Para la validación inicial, el proyecto se centra en DUOC UC, simulando un entorno académico real donde se desarrollan proyectos de software y existe la necesidad de asegurar una documentación clara, trazable y estandarizada.

En esta fase, el modelo se ajusta a los procesos y características de DUOC, sirviendo como caso piloto para:

* Probar las funcionalidades principales (mantenedores, login, permisos, plantillas).
* Evaluar la usabilidad del sistema con diferentes roles (profesores, estudiantes, administradores).
* Medir la adaptabilidad y escalabilidad del modelo.

## Escalabilidad Posterior

Una vez validado en el entorno de Duoc UC, DocuFlow está diseñado para ser replicado con éxito en cualquier otra empresa o institución. Para garantizar una implementación clara y sencilla en nuevos ámbitos, se ha definido un plan de escalabilidad que demuestra la factibilidad de aplicación a nuevas soluciones, basándose en la parametrización de sus mantenedores.

El proceso de réplica y despliegue sigue los siguientes pasos:

1. Diagnóstico y Levantamiento de Requerimientos: Se realiza un análisis inicial para comprender los procesos y flujos de trabajo de la nueva organización. Esto incluye identificar los roles de usuario, los tipos de documentos que manejan y los requisitos específicos de aprobación y validación.
2. Parametrización del Modelo: Utilizando los módulos administrativos (mantenedores) de DocuFlow, se configura el sistema para que se ajuste a las necesidades detectadas en la fase de diagnóstico. Esto permite:

* Ajustar los roles de usuario para que coincidan con la estructura de la nueva organización (ej. Jefes de Proyecto, Analistas QA, etc.).
* Adaptar las plantillas de documentación a los estándares y formatos requeridos.
* Configurar los flujos de revisión y aprobación para que se alineen con los procesos internos de control de calidad.

1. Capacitación y Despliegue Progresivo: Finalmente, se capacita al personal clave y se implementa el sistema de manera gradual. Este enfoque por fases asegura una adopción fluida y permite validar el funcionamiento del modelo con los procesos reales de la organización.

Este plan estratégico demuestra que DocuFlow no es una solución limitada a un solo contexto, sino un modelo escalable y listo para ser adaptado a las necesidades de diversas industrias, desde pymes hasta grandes corporaciones.

## Solución

La propuesta consiste en el diseño de un modelo estandarizado de documentación técnica aplicable a proyectos de software, basado en metodologías ágiles y en buenas prácticas de gestión de proyectos. A diferencia de un sistema rígido, este modelo se destaca por su flexibilidad y adaptabilidad a las necesidades específicas de cada organización.

El modelo contempla las siguientes funcionalidades clave:

* Plantillas reutilizables y validadas: Para garantizar que las plantillas iniciales sean pertinentes, su diseño se basará en una investigación preliminar sobre estándares de la industria y buenas prácticas documentales. Sin embargo, para asegurar su aprobación y uso en diferentes empresas, el modelo incorporará un mantenedor de plantillas. Esta funcionalidad permitirá a los administradores personalizar las plantillas existentes o, lo que es más importante, subir sus propios documentos específicos para que sirvan de base para la estandarización interna. Esto aborda directamente la necesidad de que cada empresa pueda utilizar sus propios formatos, haciendo que la solución sea verdaderamente útil.
* Gestión documental con permisos diferenciados: El sistema permitirá asignar permisos específicos a cada rol de usuario (administrador, desarrollador, usuario técnico, usuario no técnico), garantizando el acceso adecuado a la documentación relevante.
* Trazabilidad y control: Se integrarán funcionalidades para la carga, revisión y aprobación de documentos. Cada acción quedará registrada con el autor, la fecha y el historial de cambios, lo que asegura una trazabilidad completa y un control de versiones riguroso.
* Seguimiento visual del progreso: El modelo contará con una barra de avance documental (0% – 100%) por cada proyecto, permitiendo a los equipos visualizar de manera clara el progreso de la documentación y el cumplimiento de hitos.

## Propuestas y Alcances

El proyecto DocuFlow consiste en la implementación real de un modelo estandarizado de documentación técnica. Si bien el modelo está diseñado para ser replicable y escalable en cualquier tipo de organización, el alcance de este proyecto académico se centrará en su validación en un contexto controlado y pertinente para Duoc UC. Este enfoque nos permite demostrar la viabilidad, usabilidad y efectividad del modelo de manera concreta y medible.

### Funcionalidades Clave del Modelo

El modelo contempla una serie de funcionalidades interconectadas que aseguran la eficiencia y calidad de la documentación:

1. Acceso al Sistema (Login y Roles) Los usuarios tendrán acceso al sistema a través de credenciales personales. El modelo implementará un sistema de permisos diferenciados para controlar el acceso y las acciones de cada usuario, según su rol:

* Administrador: Posee acceso total a todas las carpetas y funcionalidades, incluyendo la aprobación final y publicación de documentos. Es el único perfil con capacidad para gestionar usuarios y configurar los mantenedores.
* Equipo de Desarrollo: Tienen permisos para crear, editar y cargar documentación técnica en las carpetas de trabajo.
* Usuarios Técnicos/No Técnicos: Su acceso es restringido, principalmente a la visualización y descarga de manuales, guías y documentos finales que les correspondan.

2. Estructuración del Modelo Documental El modelo se basa en un repositorio organizado por carpetas y plantillas. Se utilizarán herramientas colaborativas como Confluence y Notion para una gestión centralizada y accesible.

* Plantillas dinámicas: Se desarrollará un conjunto de plantillas iniciales basadas en buenas prácticas y estándares de la industria (ej. para requerimientos, especificaciones, manuales de usuario). No obstante, el sistema permitirá al administrador subir, modificar y estandarizar las plantillas propias de cada empresa, garantizando la flexibilidad del modelo.
* Formato uniforme: Cada documento seguirá un formato uniforme para mantener la claridad y consistencia visual en todos los proyectos.

3. Gestión y Trazabilidad de Documentos El sistema integra un flujo de trabajo claro para el manejo de la documentación, asegurando la trazabilidad de cada acción.

* Carga y revisión: Los usuarios podrán subir archivos para su revisión por parte de un superior o el administrador.
* Aprobación y publicación: El administrador podrá revisar, aprobar, rechazar o solicitar correcciones a los documentos. Una vez aprobados, se autoriza su publicación en el repositorio final.
* Historial de versiones: Cada documento tendrá un historial de versiones detallado que mostrará el autor, la fecha y las modificaciones realizadas, lo que permite rastrear cada cambio.
* Descarga de formatos oficiales: Para mantener la consistencia, el sistema permitirá la descarga de los formatos oficiales y las plantillas aprobadas.

4. Control de Permisos y Seguridad Se implementarán medidas de seguridad robustas para proteger la integridad de la información.

* Acceso limitado: Los accesos estarán estrictamente limitados según el rol, evitando que los usuarios accedan a información sensible o no relacionada con sus tareas.
* Control de carpetas críticas: La definición de una "carpeta crítica" es configurable por el administrador del sistema. Estas carpetas contienen documentos de aprobación final o registros históricos, y solo el administrador tiene la autoridad para mover o eliminar su contenido, protegiendo así la integridad del proyecto.
* Registro de auditoría: El sistema registrará de forma automática cada actividad, incluyendo quién subió, revisó o aprobó un archivo, creando un registro de auditoría completo para control y trazabilidad.

5. Seguimiento del Progreso Documental El modelo incorpora herramientas visuales de gestión para monitorear el avance de la documentación.

* Barra de avance: Cada proyecto contará con una barra de avance (0% – 100%) que permitirá visualizar de forma clara el progreso documental.
* Hitos de documentación: Se establecerán hitos específicos (ej. requerimientos completos, manuales validados), que se actualizarán en la barra de progreso al ser completados.
* Evaluación de estado: El modelo permitirá evaluar el estado de la documentación en relación con el cronograma del proyecto, identificando posibles desviaciones.

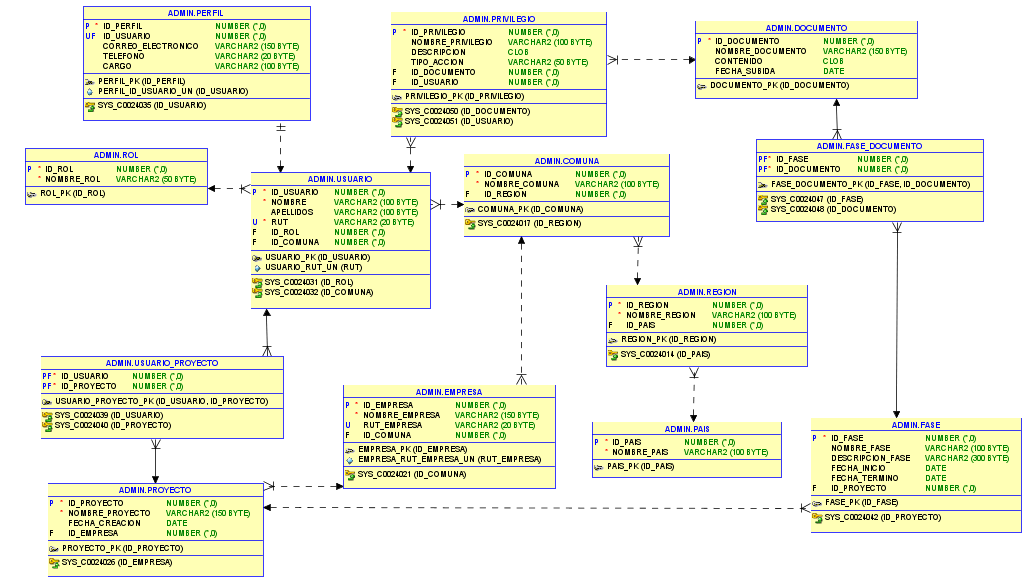
6. Validación del Modelo La validación se llevará a cabo a través de un proyecto de software real desarrollado en el marco académico, en el que se aplicarán todas las funcionalidades del modelo. Esto permitirá:

* Observar la usabilidad y la experiencia del usuario.
* Evaluar la adaptabilidad de las plantillas y permisos.
* Demostrar la escalabilidad a nivel de funcionalidades.

Los resultados de esta validación servirán para retroalimentar y ajustar los parámetros y flujos del modelo, preparándolo para una eventual réplica en otros entornos.

## Diagramas

### Diagrama de BD



Diccionario de Datos.  
1. País

* Guarda el nombre de cada país donde se desarrollan actividades.
* Ejemplo: Chile, Argentina, Perú.

### 2. Región

* Guarda las regiones dentro de cada país.
* Está conectada al país al que pertenece.
* Ejemplo: Región Metropolitana, Región de Valparaíso.

### 3. Comuna

* Guarda las comunas dentro de cada región.
* Está conectada a la región correspondiente.
* Ejemplo: La Pintana, Maipú, Puente Alto.

### 4. Empresa

* Guarda datos de las empresas que participan en proyectos.
* Incluye su nombre, RUT y comuna donde están ubicadas.

### 5. Proyecto

* Guarda los proyectos que realiza cada empresa.
* Incluye el nombre del proyecto, la fecha en que se creó y qué empresa lo gestiona.

### 6. Rol

* Define el tipo de persona en el sistema.
* Ejemplo: Alumno, Profesor, Director.

### 7. Usuario

* Guarda los datos básicos de cada persona.
* Incluye nombre, apellidos, RUT, rol que cumple y comuna donde vive o trabaja.

### 8. Perfil

* Guarda información adicional del usuario.
* Incluye su correo electrónico, teléfono y el cargo que ocupa (por ejemplo: Coordinador, Supervisor).

### 9. Usuario\_Proyecto

* Indica qué personas están participando en qué proyectos.
* Permite que una persona esté en varios proyectos y que un proyecto tenga muchas personas.

## 

### 10. Fase

* Guarda las distintas etapas o fases de un proyecto.
* Incluye el nombre de la fase, una descripción, y las fechas de inicio y término.

### 11. Documento

* Guarda los archivos o textos importantes del sistema.
* Incluye el nombre del documento, su contenido y la fecha en que fue subido.

### 12. Fase\_Documento

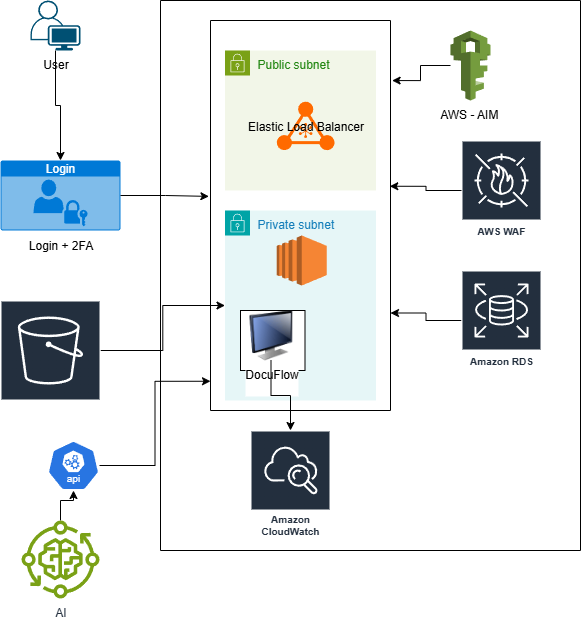
* Relaciona los documentos con las fases del proyecto.
* Permite saber qué documentos pertenecen a qué etapa.

### 13. Privilegio

* Define qué puede hacer cada persona con un documento.
* Por ejemplo: subirlo, descargarlo, modificarlo.
* Está conectado al documento y al usuario que tiene ese permiso.

### Diagrama de Arquitectura

Esperando retroalimentación de la última clase.

****

Este diagrama representa la arquitectura de DocuFlow, una aplicación segura y escalable desplegada en Amazon Web Services (AWS).

Seguridad de acceso

Login + 2FA: Se implementa un sistema de autenticación con doble factor para garantizar que solo usuarios autorizados accedan a la plataforma. Esto protege contra accesos indebidos y refuerza la integridad del sistema.

Infraestructura en la nube

Elastic Load Balancer (ELB): Distribuye el tráfico entre los servidores disponibles, asegurando alta disponibilidad y evitando sobrecargas.

VPC con subred pública y privada: Permite separar los componentes expuestos a internet (como el ELB) de los internos (como la aplicación y la base de datos), reforzando la seguridad y el control de acceso.

Aplicación central

DocuFlow: Es el núcleo funcional del sistema. Gestiona usuarios, documentos, permisos y flujos de trabajo. Se aloja en la subred privada para proteger su lógica de negocio.

Servicios de seguridad

AWS IAM (Identity and Access Management): Controla los permisos y roles dentro del entorno AWS, asegurando que cada usuario tenga acceso solo a lo que le corresponde.

AWS WAF (Web Application Firewall): Protege la aplicación contra ataques web comunes como inyecciones SQL o scripts maliciosos.

Almacenamiento y persistencia

Amazon RDS: Base de datos relacional que almacena la información estructurada del sistema (usuarios, proyectos, documentos).

Storage Bucket (S3): Repositorio de archivos como documentos, imágenes o respaldos, con alta escalabilidad y seguridad.

Comunicación e integración

API: Permite que otros sistemas o módulos se comuniquen con DocuFlow, facilitando la integración y la interoperabilidad.

Inteligencia Artificial (IA)

Aunque aparece en el diagrama, la IA no está incluida en el alcance actual del proyecto. Según el documento oficial, se considera una mejora futura. Su función potencial sería asistir en la redacción, revisión y optimización de documentos mediante generación automática de contenido y sugerencias inteligentes. Esta propuesta está descrita en la sección “Propuesta de Valor y Factor Diferenciador”.

## Propuesta de Valor y Factor Diferenciador

El modelo DocuFlow ofrece un valor agregado al no ser solo un esquema estandarizado de documentación, sino un ecosistema que potencia la eficiencia del equipo. La propuesta de valor se centra en tres pilares: estandarización, colaboración y asistencia inteligente.

Si bien la integración de un asistente basado en Inteligencia Artificial es una de las funcionalidades más prometedoras de la solución, para el alcance de este proyecto académico, se considera una futura mejora. Su inclusión en el diseño conceptual se basa en un análisis de su potencial para:

* Agilizar la creación de documentos al generar borradores automáticos a partir de datos estructurados, lo que reduce la "hoja en blanco".
* Reducir errores y omisiones al sugerir contenido y formatos basados en las plantillas y el contexto del proyecto.
* Aumentar la productividad del equipo al automatizar tareas repetitivas y ofrecer asistencia contextual durante el proceso de documentación.

El principal factor diferenciador de DocuFlow es su integración holística. A diferencia de otros sistemas de gestión documental que solo ofrecen repositorios o plantillas estáticas, DocuFlow combina de manera sinérgica:

* Estandarización: A través de plantillas configurables y una estructura de permisos y repositorios uniforme.
* Colaboración: La solución facilita el trabajo en equipo mediante el uso de herramientas asíncronas, como Google Docs o Confluence. Para el alcance del proyecto, la colaboración se gestiona a través de los flujos de trabajo (carga, revisión y aprobación), sin implicar funcionalidades de edición simultánea o en tiempo real directamente en la plataforma.
* Asistencia Inteligente: La integración conceptual de IA como futura mejora que potenciará la creación, revisión y optimización de la documentación.

## Arquitectura y Tecnologías del Proyecto

Se ha definido una arquitectura robusta y un conjunto de tecnologías con un propósito claro, garantizando que el diseño sea escalable, seguro y fácil de mantener. Se ha optado por un enfoque agnóstico de proveedor (multi-cloud) para demostrar la portabilidad de la solución.

1. Infraestructura Cloud

* Propósito: Garantizar alta disponibilidad, escalabilidad y accesibilidad remota.
* Tecnologías: Se utilizarán servicios equivalentes en AWS, Google Cloud o Azure (ej. EC2/Compute Engine/VM para la ejecución de la aplicación; S3/Cloud Storage/Blob Storage para el repositorio de documentos; y RDS/Cloud SQL/Azure SQL para la base de datos relacional). La elección de un enfoque agnóstico permite la flexibilidad de despliegue en cualquier entorno.

2. Tecnologías de Aplicación

* Backend (Python/Django): Python fue seleccionado por su simplicidad y versatilidad, mientras que Django se eligió por su robusto enfoque de "baterías incluidas", que acelera el desarrollo de APIs y funcionalidades de administración.
* Frontend y Backend (HTML/CSS/JavaScript): Se utilizarán estas tecnologías estándar de desarrollo web para asegurar la compatibilidad universal con cualquier navegador y dispositivo.
* IA (Futura Mejora): La futura integración de IA podría utilizar APIs de grandes modelos de lenguaje como OpenAI API, Vertex AI de Google Cloud o AWS Bedrock para ofrecer funcionalidades de redacción y sugerencia de contenido.

3. Base de Datos

* Base de Datos Relacional (PostgreSQL / MySQL): Se escogieron por su fiabilidad, robustez y madurez. Son ideales para la gestión de datos estructurados y relaciones complejas, como usuarios, permisos y la estructura de los proyectos.
* Base de Datos NoSQL (DynamoDB / Firestore): Serán utilizadas para el manejo de datos no estructurados, como los logs de actividad y eventos de auditoría, que requieren una alta velocidad de escritura y escalabilidad masiva.

4. Seguridad

* HTTPS/TLS 1.3: Para garantizar la encriptación de los datos en tránsito y proteger la comunicación entre el usuario y el servidor.
* IAM (Identity and Access Management): Para implementar el control de acceso diferenciado a nivel de la infraestructura y los servicios, garantizando que cada rol tenga solo los permisos necesarios.
* WAF (Web Application Firewall): Para filtrar el tráfico malicioso y proteger la aplicación de ataques comunes como la inyección SQL o el cross-site scripting (XSS).
* Encriptación AES-256: Para asegurar que los datos almacenados en reposo (en la base de datos y en los repositorios de documentos) estén protegidos en caso de un acceso no autorizado.
* Auditoría y Trazabilidad: Los logs de auditoría y el historial de versiones de los documentos permitirán un seguimiento completo de cada acción realizada en el sistema.

## Arquitectura de tres capas

El proyecto DocuFlow utilizará una arquitectura de tres capas. Este modelo es ideal para el alcance del proyecto, ya que separa las responsabilidades de cada componente de manera clara y eficiente, lo que facilita el desarrollo, la mantención y la escalabilidad futura.

### Capas de la Arquitectura

Capa de Presentación (Frontend): Es la interfaz de usuario, lo que el usuario ve e interactúa en el navegador. Se desarrollará con tecnologías estándar de la web como HTML, CSS y JavaScript para asegurar la compatibilidad y la accesibilidad. Su única función es mostrar información y enviar solicitudes a la capa de lógica de negocio.

Capa de Lógica de Negocio (Backend): Actúa como el cerebro de la aplicación. Aquí residen todas las reglas de negocio, la gestión de permisos, los flujos de trabajo (aprobación, revisión) y la integración con las herramientas de IA. Se construirá con Python (Django), exponiendo una API que la capa de presentación consume para solicitar y enviar datos.

Capa de Datos: Es la responsable de almacenar y gestionar toda la información de la aplicación. Se utilizarán bases de datos relacionales (PostgreSQL/MySQL) para los datos estructurados (usuarios, documentos, proyectos) y bases de datos NoSQL para los datos de registro (logs y eventos de auditoría), lo que optimiza el rendimiento y la escalabilidad.

### Despliegue en la Nube

Toda esta arquitectura se alojará en una infraestructura de nube (AWS, GCP o Azure). Esto garantiza que el sistema sea accesible desde cualquier lugar, tenga alta disponibilidad y pueda escalar dinámicamente según la carga de trabajo, lo que es clave para su viabilidad y replicabilidad. Si bien el prototipo se centra en esta arquitectura, su diseño modular con una API bien definida sienta las bases para una futura migración a una arquitectura de microservicios.

## Cronograma DocuFlow

La gestión del proyecto se realizará bajo una metodología ágil, dividida en sprints de dos semanas. El seguimiento de las tareas, el avance y la asignación de responsabilidades se documentará en una herramienta de gestión de proyectos como Jira, lo que permitirá una trazabilidad completa del trabajo.

1. Sprint 1: Planificación y Diseño (9 - 22 septiembre)

Objetivos: Sentar las bases conceptuales y de planificación del proyecto.

Tareas:

* Definir y validar el alcance final del proyecto.
* Realizar el levantamiento de requerimientos funcionales y no funcionales (casos de uso, historias de usuario).
* Diseñar la arquitectura del sistema y el diagrama de base de datos (primera versión).
* Elaborar el primer borrador del modelo de documentación y las plantillas base.
* Entregables:
* Documento de Alcance y Requerimientos.
* Diagrama de Arquitectura de 3 Capas.
* Diagrama de Base de Datos.
* Definición de las plantillas iniciales.

1. Sprint 2: Prototipado y Configuración (23 septiembre - 6 octubre)

Objetivos: Crear las primeras funcionalidades clave del sistema y validar el diseño.

Tareas:

* Diseñar el mockup (prototipo) de la interfaz de usuario para el login, dashboard y menú principal.
* Implementar el prototipo funcional del módulo de login con roles y autenticación.
* Desarrollar la lógica inicial del módulo de mantenedores para la parametrización de usuarios y roles.
* Configurar la integración con un repositorio de archivos en la nube.
* Entregables:
* Mockups de interfaz de usuario (UI/UX).
* Prototipo funcional del login.
* Módulo de mantenedores (versión inicial).
* Conexión exitosa con el repositorio en la nube.

1. Sprint 3: Funcionalidades Core (7 - 20 octubre)

Objetivos: Desarrollar los módulos centrales de gestión documental.

Tareas:

* Implementar el módulo de gestión de documentos (carga de archivos, previsualización y descarga).
* Configurar el flujo de trabajo de revisión y aprobación de documentos.
* Desarrollar la funcionalidad de historial de versiones para registrar los cambios y sus autores.
* Implementar una barra de progreso que muestre el avance de la documentación.
* Entregables:
* Módulo de gestión de documentos.
* Funcionalidad de historial de versiones.
* Barra de avance documental.

1. Sprint 4: Refinamiento y Seguridad (21 octubre - 3 noviembre)

Objetivos: Fortalecer la seguridad y las funcionalidades existentes.

Tareas:

* Configurar las políticas de seguridad de la infraestructura (IAM, WAF, HTTPS/TLS).
* Implementar el registro de logs detallados para auditoría y trazabilidad.
* Realizar pruebas unitarias y de integración para los módulos desarrollados.
* Optimizar el rendimiento de las funcionalidades clave.
* Entregables:
* Informe de configuración de seguridad.
* Logs de auditoría funcionales.
* Informe de resultados de pruebas.

1. Sprint 5: Validación y Ajustes (4 - 17 noviembre)

Objetivos: Poner a prueba el modelo en un entorno de validación.

Tareas:

* Realizar un caso piloto en el contexto de DUOC UC, aplicando el modelo a un proyecto de software ficticio.
* Evaluar la usabilidad y la experiencia de usuario con diferentes roles (estudiantes y profesores simulados).
* Recopilar feedback para identificar mejoras en el sistema.
* Realizar ajustes finales en la interfaz de usuario y los permisos.
* Entregables:
* Informe de resultados de validación.
* Lista de mejoras identificadas.
* Sistema refinado y ajustado.

1. Sprint 6: Entrega y Documentación (18 - 30 noviembre)

Objetivos: Finalizar el proyecto y preparar la entrega.

Tareas:

* Realizar pruebas integrales para asegurar el correcto funcionamiento del sistema.
* Elaborar la documentación técnica del proyecto (manual de usuario, manual de instalación, arquitectura final).
* Preparar el informe final y la presentación para la defensa del proyecto Capstone.
* Organizar el repositorio del proyecto en GitHub.
* Entregables:
* Sistema funcional y listo para la defensa.
* Documentación técnica completa.
* Presentación e Informe Final.