



# **NORMATIVA PARA PROYECTOS EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

**Versión 3.1:**

**Abril 2025**

# 1. Introducción

Este documento tiene como propósito establecer las pautas, requisitos y normativas fundamentales para la ejecución de proyectos en el marco del Ciclo Superior de Desarrollo de Aplicaciones Multiplataforma. Se concibe como una guía esencial que busca no solo orientar, sino también estandarizar (al menos entre los distintos grupos de la clase) y potenciar el proceso de desarrollo de aplicaciones, promoviendo la disciplina, la calidad y la coherencia en cada fase del proyecto.

A lo largo de este documento, se abordarán aspectos fundamentales que van desde la planificación inicial hasta la entrega final del proyecto, haciendo énfasis en la importancia de la especificación de requisitos, el diseño, la implementación, las pruebas y la documentación como pilares indispensables para el éxito de cualquier aplicación multiplataforma.

Por último, destacar que cada uno de los puntos que se abordan durante las siguientes secciones son de obligado cumplimiento y necesarios para aprobar la asignatura de Proyectos.

## 2. Desarrollo del proyecto

Si previamente se ha cursado otro ciclo, debemos tener en cuenta que cada uno tiene un currículum diferente y, por tanto, se tienen que realizar proyectos totalmente diferentes. La detección de similitudes en proyectos entregados anteriormente, aunque sea por el mismo alumno, puede ser motivo de suspenso.

Al mismo tiempo, no se aceptará trabajo copiado de fuentes de Internet, por ello es importante las referencias bibliográficas a los recursos empleados. En caso de que el profesor considere que el porcentaje de plagio es alto, entonces podría, en cualquier momento, suspender el desarrollo. Si esto ocurre, el alumno deberá cambiar el tema del proyecto y proceder con una nueva implementación.

Este trabajo también debe incluir un historial de versiones y cambios, que deberá llevarse a cabo a través de la herramienta Git y un repositorio en GitHub. Mantener un registro de todas las versiones anteriores del software, junto con las actualizaciones y modificaciones realizadas en cada versión, es obligatorio. Este historial sirve como memoria para que puedas rastrear la evolución del proyecto y comprender las decisiones tomadas en el pasado.

La implementación de la idea debe tener un valor de originalidad y ser un buen caso de negocio. Esto quiere decir que, por un lado, debería tener potencial para que muchas personas hagan uso de vuestro proyecto. Y, por otro lado, debéis pensar cómo obtener beneficios (aunque esto sea únicamente sobre el papel se puede dedicar una sección a una simulación realista de beneficios).

Una parte importante del proyecto, que también es de obligado cumplimiento, es que este debe ser un desarrollo original. No se permite realizar una implementación a través del uso de algún tipo de CMS como por ejemplo WordPress siempre. El desarrollo implica el uso de herramientas y lenguajes de programación, así como, potencialmente, el almacenamiento de información y uso de bases de datos.

### 3. Memoria

La extensión de la memoria debe tener entre 12.000 y 14.000 palabras. Una extensión fuera del rango afectará negativamente a la nota del proyecto. Ten en cuenta que la numeración de páginas comienza a partir del Índice. Los anexos pueden ser incluidos al final del documento, justo antes de las referencias bibliográficas, y en ningún caso contarán como páginas para la extensión de la memoria.

La memoria es un documento en el que queda recogido los resultados y el desarrollo del proyecto. En el contexto de desarrollo de software, esto incluiría documentos que describen la arquitectura del software, los requisitos del sistema (ERS), el diseño de la base de datos desde el modelo E-R, manuales de usuario, y cualquier otra información relevante como diagramas de casos de uso y diagramas UML. Estos documentos actúan como una *memoria* del proyecto, proporcionando información detallada sobre cómo está construido el proyecto y cómo debe ser utilizado.

#### 3.1. Restricciones de formato en el documento

El tipo de letra debe ser una tipografía *serif*, ya que sus característicos apoyos ayudan al lector a crear una línea visual que proporciona fluidez en la lectura. En este sentido, una de las fuentes más elegantes de este tipo es Garamond, aunque puedes usar otras como Palatino, Times New Roman, etc. El interlineado será de 1.5 puntos, y puedes usar algún tipo de sangría en el documento, como por ejemplo la *sangría de primera línea*, que se caracteriza por una sangría al comienzo de cada sección de la memoria.

Por otro lado, si abordamos la bibliografía, esta debe estar en la 7ª edición del formato APA y listada por orden alfabético en lugar de por orden de aparición. En cuanto a los anexos, si los hubiere aparecerán después del trabajo futuro y antes de las referencias bibliográficas. Respecto a las figuras y las tablas que aparezcan en el documento, estas deben tener títulos, que aparecerán en los índices de figuras y tablas respectivamente.

Toda la documentación del proyecto debe ser entregada en formato PDF.

### **3.2. Tapa-cubierta**

La tapa-cubierta o portada sería como la tapa dura de un libro (es decir, únicamente es útil si alguien desea encuadernar su memoria). En la tapa-cubierta del ejemplar de la memoria deberán figurar los siguientes datos:

1. El logo del colegio.
2. El nombre de la etapa educativa.
3. El nombre del Grado Superior cursado. El título del proyecto.
4. Nombre y apellidos del alumno
5. Nombre y apellidos del profesor
6. Año

### **3.3. Portada**

Es la primera página de la memoria después de la tapa-cubierta, donde deben aparecer los mismos datos que figuran en la tapa-cubierta.

### **3.4. Contenido previo al cuerpo de la memoria**

Tras la portada y antes del cuerpo de la memoria, deberán figurar en el orden que se indica:

1. Resumen en español. Palabras clave en español.
2. Resumen en inglés.
3. Palabras clave en inglés. Agradecimientos.

4. Índice de contenido, figuras y tablas. En el índice de contenido deben aparecer los elementos mencionados anteriormente y, además, los acrónimos (siguiente punto), los capítulos (incluidos introducción, conclusiones y trabajo futuro), bibliografía y anexos.
5. Acrónimos.

## 4. Contenido del cuerpo de la memoria

Algo importante para toda la memoria es que esta se escribe de forma impersonal, usando la pasiva. Así, frases como por ejemplo: *Como vemos en esta sección...*, *Debemos hacer hincapié en...*, *Podríamos afirmar que...*, etc., deben ser escritas como: *Como se ha visto en esta sección...*, *Se debe hacer hincapié...*, *Se podría afirmar que...*

Por otro lado, la memoria puede tener diferentes secciones según el tipo de proyecto que se desarrolle, por eso se deja cierta libertad a la hora de establecer el esquema del documento. Sin embargo, existen ciertos puntos que, obligatoriamente, deberán aparecer en la memoria.

**Introducción:** justificación del tema seleccionado y resumen del contenido del trabajo.

**Justificación del tema elegido:** por qué habéis elegido la idea, cuáles son sus puntos fuertes, qué problemas habéis detectado y qué soluciones proponéis al respecto.

**Objetivos:** explicación de los objetivos que se pretenden alcanzar. Aquí se habla del objetivo general del proyecto y su finalidad, el cual resume la idea central en un sentido amplio, pero delimitando. Además, también se habla de los objetivos específicos, que detallan los procesos necesarios para la completa realización del proyecto.

**Metodología:** incluso aunque no sepas nada sobre desarrollo de proyectos, vas a terminar utilizando (consciente o inconscientemente) alguna metodología. En esta sección debes explicar qué tipo de metodología has empleado, por lo que te recomiendo que leas acerca de las metodologías de desarrollo más comunes.

**Tecnologías y herramientas utilizadas en el proyecto:** aquí detallaremos a fondo las tecnologías y herramientas utilizadas en el proyecto, así como las causas que han llevado al alumno a tal decisión (los lenguajes de programación, editores de código, SGBD, etc.).

**Despliegue y pruebas:** apartado en el que se explican algunas de las pruebas llevadas a cabo. Imagina que has desarrollado una aplicación web, pues en este apartado podrías contar

cómo has hecho el despliegue de la aplicación, qué pruebas iniciales has ejecutado, cómo has testeado la funcionalidad de la aplicación, pruebas de los casos de uso que hayas obtenido en la SRS, etc.

**Conclusiones:** detallar los objetivos que finalmente se han alcanzado en el proyecto, que no siempre debe de coincidir con los objetivos propuestos. En este caso, explicar las causas por las que no se han logrado. Comentar la opinión sobre el trabajo que se ha realizado, tanto las dificultades surgidas como el desarrollo en general. En esta sección se suele contrastar los objetivos iniciales con los resultados obtenidos.

**Trabajo futuro:** posibles mejoras o trabajos adicionales que han podido (o no) ser contemplados durante la fase de planificación. El trabajo futuro, como su propio nombre indica, hace referencia a aquellas líneas de trabajo que podrían tomarse en caso de continuar con el desarrollo del proyecto. Por esta razón, deben ser procesos concretos, viables y que se encuentren dentro del ámbito del proyecto.

De manera adicional, también puedes considerar más apartados para darle un poco más de vida a tu memoria y tratar de perfeccionarla. No son secciones obligatorias, pero te adjunto algunos puntos que podrían ser de interés:

**Estado del arte:** hacer una pequeña sección donde hayas investigado acerca del campo en el que encaja tu aplicación daría más valor y permitiría entender mucho mejor el desarrollo. Por ejemplo, supongamos que mi aplicación es una App (una aplicación que funciona dentro de la blockchain). En este caso, podría dar algunas pinceladas sobre qué es la blockchain, cómo funciona, qué son las Apps, ventajas, desventajas, utilidades, etc. Se trata de darle un poco más de contexto a tu desarrollo.

**Profundizando en los conceptos:** un anexo podría incluir una mayor profundidad técnica sobre alguno de los conceptos que estés utilizando. Me explico: imagina que en un punto de tu aplicación haces uso de un algoritmo genético para obtener un cierto resultado. Pues bien, un anexo podría explicar brevemente qué es un algoritmo genético, para qué sirve, cuál es su utilidad y cómo funciona.

**Manual de usuario:** en otro de los anexos podrías incluir un manual de usuario de la aplicación. En este manual se explicarían las diferentes partes y funciones, todo ello evitando palabras demasiado técnicas, ya que está enfocado a usuarios sin conocimientos informáticos. La experiencia de usuario es algo que se agradece en cualquier aplicación, por lo que ponerte

en la piel de uno de ellos podría ser un añadido de gran valor. En cualquier caso, esto no contaría como extensión de la memoria al ser un anexo.

En cualquier caso, sé original. Estas pautas, aún de obligado cumplimiento, dan cierta libertad de contenido. Esto no debe ser visto como un manual lleno de normas, sino más bien como una guía para facilitar el desarrollo de ideas. En tu mano queda pensar en qué quieres hacer, cómo lo quieres hacer y qué quieres contarle al resto de personas. ¿Cómo te gustaría que los demás vieran lo que has estado desarrollando durante tres meses?

## **5. Especificación de Requisitos Software**

Como ya sabemos, la realización de un proyecto, en ocasiones, implica que debemos generar una gran cantidad de documentación. Hay parte de esta documentación que está a cargo del equipo de proyecto (vosotros/as), pero otra parte debe ser elaborada por el equipo de dirección, director, patrocinador, etc.

Una Especificación de Requisitos Software (SRS, por sus siglas en inglés) es un documento formal que describe en detalle las funciones, características, y restricciones de un sistema software. Este documento sirve para varios propósitos. Por un lado, proporciona una descripción precisa y clara de lo que el software debe hacer, evitando malentendidos entre los desarrolladores y los stakeholders. Además, actúa como una guía para los desarrolladores, asegurando que el software se construya de acuerdo a las expectativas y necesidades del cliente.

Una SRS puede llegar a ser un documento muy amplio, dependiendo del proyecto. Algo que puede añadir valor a tu memoria es buscar un posible índice de SRS. Puedes investigar acerca de cómo hacer una buena SRS, sus partes o cuáles son los puntos que más te interesan. Al mismo tiempo, podrías añadir parte de estos puntos como índices de tu memoria si lo crees necesario. Toda esta información podría complementar la memoria que hagas.

## **6. Defensa del proyecto**

La memoria y el proyecto deben ser entregados, como muy tarde, tres días antes de la fecha de defensa. En dicha defensa podrá estar presente cualquier profesor disponible en ese momento. Se aconseja realizar una presentación de diapositivas para que esta sirva de apoyo durante la explicación. Al mismo tiempo, se recomienda llevar el propio portátil (sobre todo para

enseñar el funcionamiento en vivo de la aplicación) con salida HDMI. La duración máxima de la presentación será de 20 minutos, con tiempo posterior para preguntas y ruegos.

La fecha de defensa del proyecto y entrega del mismo se comunicará lo antes posible, aunque suele ser sobre la segunda semana de junio.

## **7. Criterios de evaluación**

La implementación y desarrollo del proyecto es lo que más peso tiene a la hora de evaluar, ya que será un 50 % de la nota total en la asignatura. Por otro lado, la memoria contará un 30 %. Y por último, la presentación del proyecto tendrá un peso total de 20 %.

## **8. Conclusión**

Hemos visto que este documento contiene algunas directrices en el desarrollo de proyectos para el Ciclo Superior de Desarrollo de Aplicaciones Multiplataforma. Por ello, se considera un componente esencial y permite una ejecución más eficiente y ordenada de los proyectos. Además, al preparar a los estudiantes para los desafíos del entorno laboral, no solo se busca profundizar en la teoría, sino que también se pretende inculcar habilidades prácticas esenciales para el desarrollo exitoso de aplicaciones en diferentes