

# **INSTRUCCIONES PARA REDACTAR Y PRESENTAR EL PLAN DE GESTIÓN, ANÁLISIS, DISEÑO Y MEMORIA DEL PROYECTO**

# ***ÍNDICE***

## Tabla de contenidos

1. ESTRUCTURA Y CONTENIDOS.....	2
2. REQUISITOS DE ESTILO Y FORMATO.....	5

# 1. Estructura y contenidos

La memoria debe tener las partes que se describen a continuación. Para cada una se da un breve resumen de los contenidos que debe tener.

## 1. Introducción

Resumen del proyecto, propósito, alcance, objetivos, entregables e hitos principales. Alrededor de una página es suficiente. No olvidar una breve descripción de la estructura del resto del documento.

## 2. Organización del proyecto

Equipo del proyecto: integrantes del mismo, roles y responsabilidades. Qué hace dentro del proyecto cada miembro del equipo. Aunque es normal que todo el mundo haga varias cosas, también es importante que haya responsables definidos para las tareas importantes. Es importante designar a un director o directora de proyecto.

## 3. Plan de gestión del proyecto

### 3.1 Procesos

En esta sección se describe cómo se llevarán a cabo distintas tareas que hay que realizar en distintos momentos del proyecto.

#### 3.1.1. Procesos de inicio

- Cómo se van a identificar y asignar recursos (p.ej. conseguir servidores en cloud o teléfonos móviles para pruebas, pero también registrarse para acceder a API que se quieran integrar o a herramientas online que se quieran usar etc.).
- Cómo se va a abordar la formación inicial de los miembros del equipo (revisar qué tecnologías se van a usar, qué componentes se van a integrar, con qué API hay que conectar y quiénes tienen que formarse, o autoformarse, en todas esas cosas y de qué manera (hacer algún curso online, planificar algo de tiempo para auto-formación con tutoriales y documentación etc.).

#### 3.1.2. Procesos de ejecución y control

- Cómo se llevarán a cabo las comunicaciones internas, el registro de las decisiones tomadas en reuniones, la redacción de las actas etc.
- Cómo se van a determinar las tareas a realizar y el reparto de las mismas a integrantes del equipo en el día a día.
- Cómo se abordarán los temas de gestión del equipo humano (p.ej. la resolución de disputas).
- Qué se va a hacer respecto a medidas de progreso y monitorización del estado del proyecto (qué se mira/mide, cada cuánto tiempo, qué se hace si se detectan problemas de rendimiento o avance insuficiente o desviaciones respecto al plan inicial...).
- Cómo se hará la entrega de resultados.

#### 3.1.3. Procesos técnicos

Describir los métodos, herramientas y técnicas necesarios tanto para construir el software (p.ej. herramientas de desarrollo) como para desplegarlo y probarlo (todos los necesarios para dar soporte a los planes descritos a continuación).

## 3.2. Planes

Un plan establece un objetivo y, en general, qué necesitamos para conseguirlo. Llevar a cabo los planes que se describan aquí requerirá aplicar los procesos descritos anteriormente, algunos una vez y otros muchas veces. Un plan suele incluir cuándo se llevan a cabo estos procesos (periódicamente, o en fechas concretas, o “al menos N veces” etc.) y quién es responsable (personas concretas a veces, pero generalmente roles) de hacerlos, o de asegurarse que se hacen.

### 3.2.1. Plan de gestión de configuraciones

- Convenciones de nombres (documentos) y estándares de código (guías de estilos).
- Responsable o responsables de las distintas actividades (puesta en marcha, apoyo al equipo, revisión de commits, copias de seguridad, control de las versiones entregadas a cliente...).
- Recursos: repositorios de control de versiones (cuáles, cuántos, permisos de acceso a los mismos) y sistema de gestión de incidencias.
- Procedimiento para realizar cambios al código fuente y los documentos técnicos: workflow de control de versiones utilizado, cuándo/cómo se permiten realizar commits al repositorio compartido, si tienen que ser aceptados por alguien previamente o no, qué hay que anotar en el sistema de gestión de incidencias, quién decide el estado de las incidencias, en qué estados puede estar una incidencia etc.

### 3.2.2. Plan de construcción y despliegue del software

- Cómo se construye e integra el software: si hay scripts de construcción automatizada o no (en ese caso qué se usa, y cómo se garantiza que todos los participantes compilan igual y con las mismas dependencias), qué se incluye en la construcción (descarga y actualización de dependencias, compilación, ejecución de tests automáticos...) y cada cuánto se construye (compila, integra, prueba) el sistema completo, cómo se configuran los computadores de los desarrolladores.
- Cómo se despliega el software más allá de las máquinas de desarrollo: contenedores, máquinas virtuales, servidor en cloud etc. y cómo se configuran esos entornos (rutas, usuarios y contraseñas, puertos y otros elementos).

### 3.2.3. Plan de aseguramiento de la calidad

- Guías para la documentación de diseño del software y otros documentos del proyecto, guías para el diseño gráfico de las GUI (estética, usabilidad...).
- Actividades de control de calidad del código que se realizarán: revisiones de código por pares, revisiones de requisitos o diagramas UML por pares, tipos de tests automáticos o manuales que se llevarán a cabo.

### 3.2.4. Calendario del proyecto y división del trabajo

- Diagrama de Gantt que recoja las tareas a realizar. Tened en cuenta que trabajáis con dos iteraciones y por tanto que hay una entrega intermedia y una final, y esto debe estar reflejado en este diagrama. Tened en cuenta que es normal que lo tengáis que actualizar conforme avance el proyecto (cuándo y cómo establezcáis en la sección 3.1.2 (Procesos de ejecución y control)).
  - Debe quedar claro qué requisitos van a estar completados en la primera iteración y cuáles en la segunda. Es posible que para la primera iteración no se planifique completar ningún requisito, pero en ese caso tiene que planificarse qué se hará y que faltará por hacer para cada requisito.
- División del trabajo en partes (los módulos del software a desarrollar, pero también la documentación, el diseño gráfico, instalaciones o despliegues, pruebas manuales etc.) y reparto de los mismos entre el equipo de desarrollo, al menos a alto nivel (el reparto de labores concretas en el día a día no se detalla aquí, pero hay que explicar bajo qué criterios y

quién/cómo se hace eso en la sección 3.1.2 (Procesos de ejecución y control). Debe haber una correspondencia con las tareas que aparecen en el diagrama de Gantt (que no necesariamente tiene que ser una relación uno a uno).

- Verificad que esta división del trabajo cubre todos los requisitos.

## 4. Análisis y diseño del sistema

### 4.1 Análisis de requisitos

Hay que completar y detallar los requisitos preliminares incluidos en la propuesta técnica y económica. Recordad que los requisitos deben ser completos, concretos, medibles cuando tenga sentido y lo menos ambiguos posible. También es importante que estén identificados para facilitar su trazabilidad.

### 4.2. Diseño del sistema

- Diagramas arquitecturales (de módulos, de componentes y conectores, de distribución), patrones de diseño y estilos arquitecturales que se aplicarán. Las interfaces (de módulos y de componentes) son especialmente importantes. También lo son los protocolos de comunicación entre componentes.
- Tecnologías elegidas (lenguajes de programación, componentes que se integrarán, API web externas con las que se conectará etc.).
- Otros aspectos técnicos de interés (p.ej. si hay base de datos si va a ser SQL o NoSQL, si hay una API Web va a ser REST[ful] o no, si algunas de las operaciones van a ser asíncronas o no, si va a ser una aplicación móvil o de escritorio será nativa o se van a usar tecnologías web, cómo se van a considerar los requisitos de seguridad o de prestaciones, cómo y dónde se harán las instalaciones y despliegues etc.)

Hay que justificar todas las decisiones de diseño. Esto exige contestar a dos preguntas sobre cada decisión: ¿qué alternativas se barajaron? y ¿por qué se eligió una y no las otras?

## 5. Memoria del proyecto

**Este capítulo no se rellena en la primera entrega.**

En este capítulo se describirá cómo se ha llevado a cabo el proyecto, qué cambios se han hecho respecto a la versión inicial, imprevistos surgidos, etc.

### 5.1. Inicio del proyecto

Describir cómo transcurrió esta fase del proyecto, especialmente los resultados de llevar a cabo los procesos descritos en la sección 3.1.1 (Procesos de inicio).

### 5.2. Ejecución y control del proyecto

Describir cómo transcurrió esta fase del proyecto, especialmente los resultados de llevar a cabo los procesos descritos en la sección 3.1.2 (Procesos de ejecución y control) y en la sección 3.1.3 (Procesos técnicos). No olvidar:

- Cómo se ha realizado el reparto de trabajo entre miembros del equipo. Cómo ha transcurrido la comunicación interna.
- Cómo se ha medido el progreso del proyecto. Cómo se sabía el trabajo realizado, el trabajo pendiente y lo que estaba haciendo cada persona.
- Los ajustes realizados cuando se detectaron divergencias frente al calendario inicial (ajustes en el trabajo y/o ajustes en el calendario). Si se han identificado las causas de estas divergencias, explicarlas.
- Adecuación de las herramientas y tecnologías empleadas. Si ha habido que cambiar alguna decisión de diseño o tecnología, y por qué.

- Funcionamiento de los procesos de control de versiones del código, construcción y despliegue. ¿Ha habido problemas con las integraciones? ¿Problemas con los despliegues? ¿Se han perdido cosas por errores humanos? ¿Cómo se han abordado estas tareas?
- Pruebas del software. ¿Se han podido cumplir las ideas que se tenían al respecto?

### 5.3. Cierre del proyecto

Al menos:

- Comparar las estimaciones iniciales (tamaño, esfuerzos, costes) con los resultados finales, analizar los resultados y tratar de expresar algunas lecciones aprendidas.
- Lecciones aprendidas sobre herramientas y tecnologías.
- Recopilar los esfuerzos dedicados al proyecto por cada uno de los participantes: horas trabajadas y actividades realizadas por cada persona.

## 6. Conclusiones

**Este capítulo solo se rellena en la entrega final.**

Además de conclusiones personales (razonadas) sobre el transcurso del proyecto realizado, es importante plantear ideas para mejorar los procesos llevados a cabo: si hubiera que iniciar un nuevo proyecto inmediatamente usando una metodología de gestión basada en procesos, ¿qué cambios haríais respecto a los procesos que habéis seguido durante este proyecto? ¿Qué cosas está claro que haríais de otra forma? ¿Qué cosas seguiríais haciendo más o menos igual?

## Glosario

### Anexo I. Actas de todas las reuniones realizadas

**Este anexo solo se rellena en la entrega final.**

Incluir aquí todas las actas de las reuniones realizadas, ordenadas cronológicamente.

Se pueden incluir otros apéndices o anexos si es necesario.

## 2. Requisitos de estilo y formato

Dedicaremos varias clases de teoría (unas 3 horas) a hablar sobre la escritura de documentos técnicos. Cualquier cosa sobre lo que pone aquí que pueda no quedar clara en una primera lectura, debería estar clara después de que veamos ese tema en clase de teoría.

- La ortografía y la gramática son correctas.
- El estilo de escritura es adecuado para un documento técnico.
- No hay un límite de páginas o palabras, pero se ha hecho un esfuerzo para expresar todas las ideas fundamentales sin texto superfluo e innecesario.
- Se ha hecho un esfuerzo para usar adecuadamente un lenguaje inclusivo.
- Sobre el formato:
  - Páginas de tamaño A4.
  - El tamaño de la tipografía del texto normal está entre 10 y 12 puntos.
  - Márgenes de entre 2,5 y 3,5 cm a cada lado, y de no más de 3 cm arriba y abajo.
  - Figuras y tablas centradas.
  - Todo el texto justificado (alineado a izquierda y derecha).
- La estructura del documento (capítulos, secciones, subsecciones...) es la indicada en el capítulo 1 y se numeran cómo se indica ahí.
  - El estilo tipográfico de los encabezados de capítulos, secciones y subsecciones es distinto al del texto normal (y distinto entre ellos, al menos en el tamaño de la fuente que debe ser mayor para capítulos que para secciones y subsecciones).
- El documento, además de la estructura indicada en el capítulo 1 debe tener además:

- Una portada con estos contenidos: un título (“Plan de gestión, análisis, diseño y memorial del proyecto X” donde X es el nombre de vuestro proyecto), el nombre y número del grupo, el nombre y NIP de cada integrante del grupo, la URL de la organización en GitHub creada para el grupo y, opcionalmente, un logotipo identificativo.
- Una tabla de contenidos (o índice general).
- Las páginas están numeradas (en la cabecera o en el pie).
- Las figuras están numeradas y todas se referencian al menos una vez desde el texto. Esa referencia incluye el número de la figura. Igual para las tablas si las hay.
- Las figuras se ven bien. Especialmente hay que asegurarse de que los diagramas con textos pequeños son legibles (si son detallados y grandes, tras hacer zoom sobre ellos siguen siendo legibles).

El documento se entregará en formato PDF. Deberá tener una tabla de contenidos interactiva que permita navegar haciendo clic a todos los capítulos, secciones etc. del documento.