

# PROYECTO

# Desarrollo de Aplicaciones Web



ESCUELA  
INTERNACIONAL  
DE **GERENCIA**

The logo for 'eig' is written in a bold, lowercase, sans-serif font. The letters are a deep purple color. The 'e' is a simple rounded shape, the 'i' has a small dot, and the 'g' has a long, curved tail that extends downwards and to the right.

# TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES WEB

## MÓDULO: PROYECTO DE DESARROLLO DE APLICACIONES WEB



# ÍNDICE

OBJETIVO .....	5
JUSTIFICACIÓN .....	6
ANÁLISIS DE LA COMPETENCIA, SI EXISTE.....	7
PROPUESTA DETALLADA .....	9
Sketch.....	9
Wireframe.....	10
Mockup.....	11
Prototipo .....	12
PLANIFICACIÓN TEMPORAL .....	13
Especificación de los objetivos de las iteraciones .....	15
Prototipado en cada iteración .....	16
DISEÑO DE LA APLICACIÓN .....	17
Esquema funcional.....	17
Diseño de las estructuras de datos.....	17
Estudio de la interacción del usuario con la aplicación .....	17
Guía de estilo.....	17
Vídeo demostrativo.....	20
CODIFICACIÓN.....	21
Lenguajes, herramientas y metodologías empleadas.....	21
Aspectos destacables de la implementación .....	21
Uso de estándares y normas de accesibilidad.....	21
DESPLIEGUE DE LA APLICACIÓN.....	22
EVALUACIÓN Y PRUEBAS .....	23
MANUAL DE USUARIO .....	24
Descripción de la aplicación.....	24
Funcionalidades y características .....	24
CONCLUSIONES .....	25



## OBJETIVO

En este apartado debes exponer cuál es el objetivo principal de tu proyecto. Considera que es una explicación sencilla de la funcionalidad que quieres conseguir con tu aplicación.

Los objetivos son declaraciones escritas sobre los resultados que queremos alcanzar en un tiempo determinado. Los objetivos son, por lo tanto, los fines hacia los que está orientada la actividad del proyecto. La definición de los objetivos es uno de los pasos para crear proyectos en el que se debe realizar una planificación.



Al determinar los objetivos del proyecto tenemos que distinguir entre **objetivos principales o generales** y **objetivos específicos**.

- **Los objetivos generales** indican el propósito central del proyecto, por lo que debemos considerarlos a largo plazo.
- **Los objetivos específicos** son pasos que hay que ir dando para consolidar la consecución de los objetivos generales hacia los que apunta el proyecto.

Los objetivos generales se van “afinando” durante el proceso de desarrollo. Al principio debemos evitar partir de ideas vagas e imprecisas, aunque también debemos considerar estos objetivos se irán concretando durante el proceso. Al avanzar en la definición del problema y sus soluciones, los objetivos se irán ajustando, se irán definiendo claramente, evitando de esta manera que los objetivos generales sean imposibles o muy difíciles de alcanzar.

## JUSTIFICACIÓN

En este apartado debes explicar cuáles son las **motivaciones** que te llevan a desarrollar la aplicación en concreto que has elegido. Puedes incluir **factores personales, sociales, situaciones de mercado, etc...**

Podrías plantear qué utilidad tendrá, en general. Por ejemplo, en base a alguno de estos supuestos:

- Si se soluciona una necesidad o problema existente
- Si tendrá algún tipo de impacto social o ayudará a las personas en algún modo
- Si creará alguna forma nueva de hacer las cosas
- Si planteará algunas opciones que antes no existían
- Si habrá grupos de usuarios concretos que se beneficien de tu proyecto especialmente



## ANÁLISIS DE LA COMPETENCIA, SI EXISTE

Los nuevos emprendedores tecnológicos responden principalmente a dos grandes perfiles, los que son técnicos o informáticos y los que no lo son. Como el mercado de aplicaciones para web o smartphone es de reciente creación, es complicado encontrar expertos en dicho campo. Esto hace que los perfiles más técnicos, más concentrados en perfeccionar el producto que en venderlo, a veces sufran algunas dificultades, y que los nuevos emprendedores, venidos de múltiples sectores, muchas veces no entiendan en profundidad la temática de Internet y los nuevos mercados tecnológicos.



En el mercado de las apps **existen varios escenarios de competencia**, y debemos prestarles la debida atención a todos y cada uno de ellos. Si no lo hacemos, nuestras probabilidades de conseguir un éxito lo suficientemente grande se reducen drásticamente.

- **El primer escenario de competencia puede definirse como la competencia clásica**, esa que nos ofrecen los productos sustitutos y los productos de la competencia que están enfocados a cubrir la misma necesidad que nuestro producto. Debemos prestar especial atención tanto a otras aplicaciones que resuelvan el mismo problema de forma similar o distinta, pero, al mismo tiempo, es de vital importancia analizar las posibles soluciones que pueden encontrar los usuarios para cubrir su necesidad en el mundo offline. Y es que es necesario ser consciente que el usuario puede acceder también a soluciones offline que le solucionen el problema.

**Es a este escenario al que mayor atención le prestaremos construyendo un informe que refleje con detalle el estado de esa competencia, si existe.**

Los siguientes escenarios son relevantes una vez desarrollada la aplicación, pero deberemos tenerlos en cuenta de forma previa a este hecho para poder anticipar nuestras necesidades futuras y considerarlas en el proyecto.

- **El segundo escenario es la competencia para conseguir una descarga, en el caso de una aplicación para Smartphone o una visita, en el caso de una aplicación web.**

Obviamente la manera más fácil de conseguir el éxito en este escenario es salir en la lista de Tops más descargados de los distintos Stores, en el caso de una aplicación compatible para móviles. Sin embargo, para hacerse un hueco en este tipo de listas va a ser necesario que tengamos varios miles de descargas diarias. Una buena manera de afrontar este reto es ser realista y asumir que muy probablemente no apareceremos inicialmente en dicha lista. En este caso lo preciso es explotar otros métodos, como conseguir menciones en blogs y foros que enlacen a nuestro sitio o a nuestra ficha en los principales Stores. Los códigos QR o incluso la publicidad pueden ser buenas opciones para empujar nuestras descargas en los inicios. Lo mismo puede considerarse para una aplicación web, en cuyo caso el objetivo serían visitas a la misma en lugar de descargas.

En el caso de una aplicación con versión para teléfono móvil, una vez en el teléfono, lo difícil es mantenerse. Los usuarios tienen tendencia a borrar las aplicaciones que no han usado en un tiempo, o incluso las que acaban de utilizar, pero no han respondido a sus expectativas. Además, muchos dispositivos Android padecen de tener poca memoria interna para apps, y no todas se pueden trasladar a la tarjeta SD. La conclusión es clara, más vale conseguir que usen nuestra aplicación, no hacerla muy pesada y en caso de no conseguirlo, que al menos se pueda trasladar a la tarjeta de memoria. Si hablamos de una aplicación web, deberemos considerar como conseguir que el usuario vuelva a visitarnos tras usar nuestra aplicación.

- **Entrando en el terreno del uso de la aplicación, debemos darnos cuenta que este es el tercer escenario de competencia.** El tiempo dedicado a su Smartphone por parte del usuario es limitado y difiere según la cultura. Lo mismo puede aplicarse al tiempo de navegación por Internet donde encontraría nuestra aplicación. Al ser un recurso finito, el usuario va a dedicarlo a aquellas aplicaciones que maximicen su bienestar, y nuestro objetivo debe ser que nos podamos contar entre ellas. Algunos trucos pueden ser la utilización de técnicas de gamificación para aumentar el compromiso o engagement del usuario, o conseguir solucionar de manera eficiente el problema que tenía éste y que le hizo llegar hasta nuestra aplicación. En todo caso, al menos debemos presentar un entorno agradable en la aplicación, dedicando nuestros esfuerzos a mejorar la experiencia de usuario (UX).



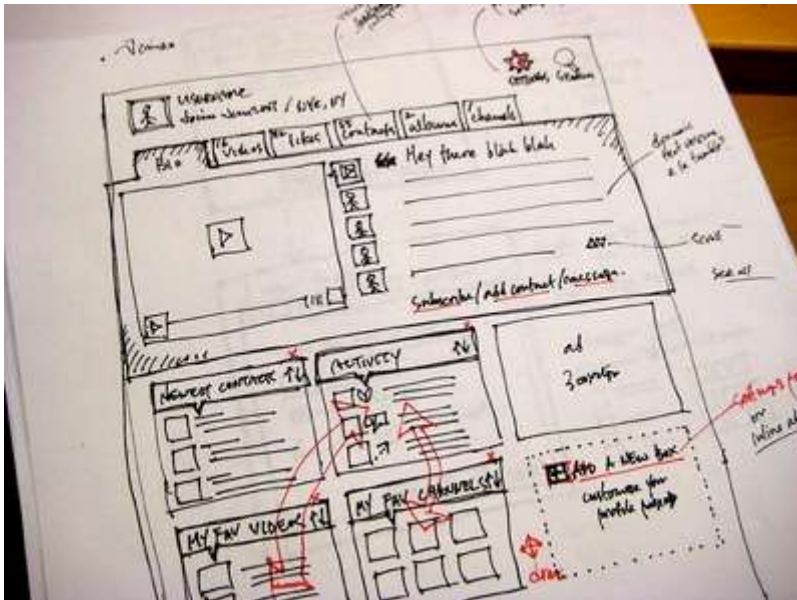
## PROPUESTA DETALLADA

En este apartado **deberás explicar en profundidad cómo es el funcionamiento de la aplicación que vas a desarrollar**. Debes centrarte en la aplicación final, una vez que esta esté completamente desarrollada, no en los prototipos intermedios de su desarrollo.



Además de la explicación de su funcionamiento debes presentar los modelos provenientes de las diferentes fases de diseño de la aplicación:

### Sketch

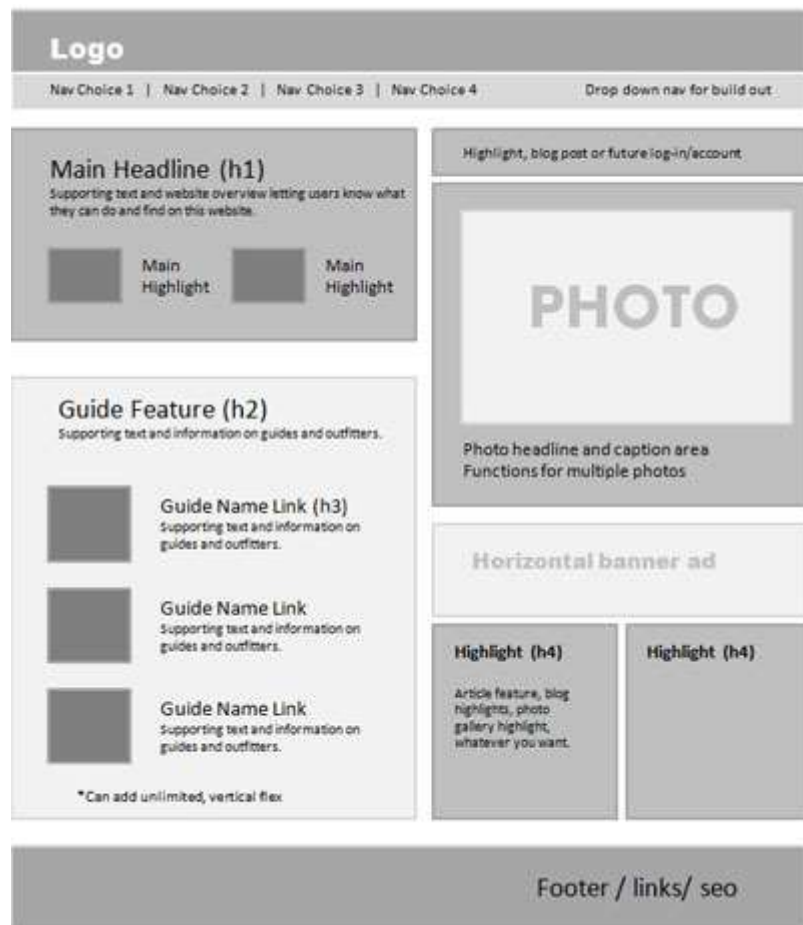


*Es un bosquejo estático en baja calidad de un diseño.*

Un sketch es un dibujo rápido o bosquejo de guía general que no tiene por qué tener muchos detalles y que reproduce un concepto, idea, o generalidad de un proyecto de una manera muy sencilla. Por generalidad se realiza a mano con lápiz y borrador por su facilidad en el momento de plasmar una base o punto de partida.

El entregable es una imagen, que habitualmente será la fotografía de un papel. Irá incluida en el documento a entregar.

## Wireframe



*Es una representación estática en baja calidad de un diseño.*

En esta representación se definen para una fácil comprensión:

- **¿Qué?:** Los principales grupos de contenido
- **¿Dónde?:** La estructura de la información
- **¿Cómo?:** La descripción y visualización básica del usuario e interacción de la interfaz

Los Wireframes no son sólo un conjunto de cuadros grises, aunque puedan verse exactamente así. **Considera a el Wireframe como la columna vertebral de tu diseño donde aparecerá una representación de cada parte importante del producto final.**

El entregable es una imagen. Irá incluida en el documento a entregar.

## Mockup



*Representación estática de un diseño en calidad media o alta*

El mockup representa la estructura de la información, visualiza el contenido y demuestra las funcionalidades básicas de una manera estática y permite revisar la parte visual real del proyecto.

El entregable es una imagen. Irá incluida en el documento a entregar.

## Prototipo



*Representación navegable del producto final. La calidad pueda variar entre media y alta.*

**El prototipo simula la interacción interfaz de usuario** y debe permitir que el usuario:

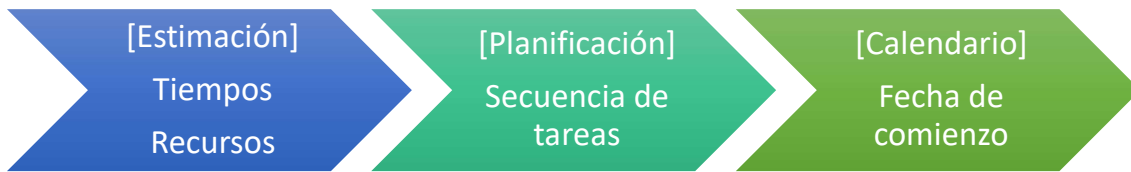
- Experimente interactuando con la interfaz y contenido del proyecto.
- Pruebe las principales interacciones de una manera similar al producto final.

El entregable es un HTML, PPT, Animación o formato navegable.

# PLANIFICACIÓN TEMPORAL

La **planificación temporal** consiste en la **identificación de tareas, la asignación de tiempos y recursos a dichas tareas y la planificación de la secuencia de ejecución de forma que el tiempo de desarrollo del proyecto sea mínimo.**

El objetivo del gestor del proyecto es definir todas las tareas del proyecto, identificar las que son críticas y hacerles un seguimiento para detectar de inmediato posibles retrasos. La planificación temporal distribuye el esfuerzo estimado a lo largo de la duración prevista del proyecto.



Los Principios de la planificación temporal son:

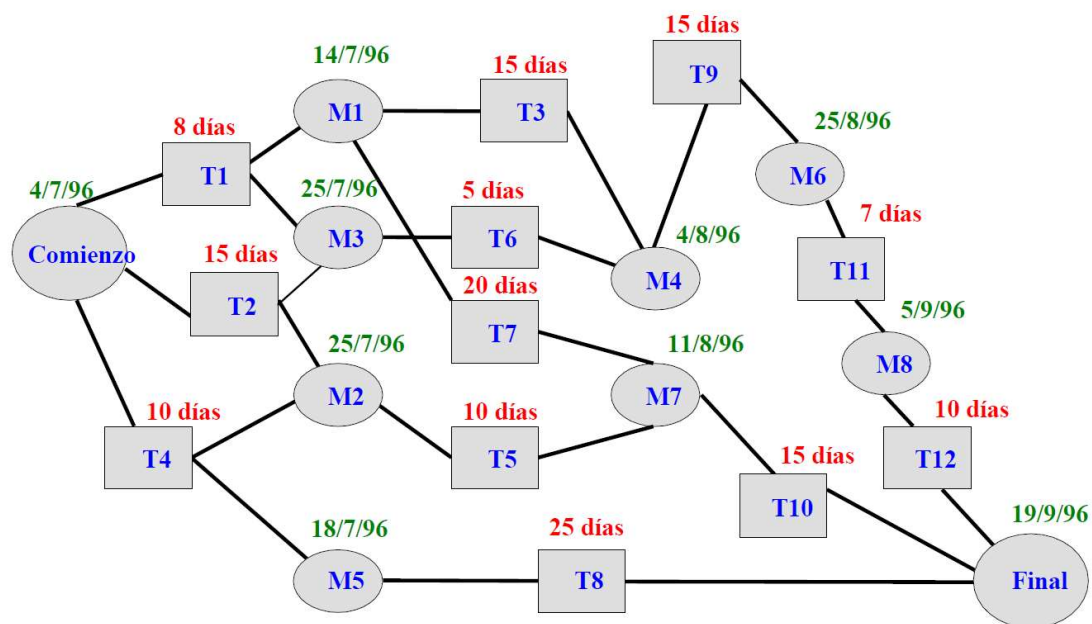
- **Compartimentación:** descomposición del proyecto en un número manejable de tareas.
- **Interdependencia:** Se deben determinar las dependencias de cada tarea.
- **Asignación de tiempo:** A cada tarea se le debe asignar un cierto número de unidades de trabajo, una fecha de inicio y otra de finalización.
- **Validación del esfuerzo:** A medida que se realiza la asignación de tiempo, el gestor del proyecto se tiene que asegurar de que los técnicos necesarios estarán disponibles en cada momento.
- **Responsabilidades definidas:** Cada tarea que se programe debe asignarse a un miembro específico del proyecto. En este caso sólo habrá un miembro en el proyecto.
- **Resultados definidos:** El resultado de cada tarea, normalmente un producto, deberá estar definido. Los productos se combinan generalmente en entregas.
- **Hitos definidos:** Todas las tareas grupos de tareas deben asociarse con algún hito del proyecto. Se considera un hito cuando se ha revisado la calidad de uno o más productos y se han aceptado.

Para nuestro proyecto **usaremos el modelo de red de tareas para la planificación temporal** del mismo. **La red de tareas es una representación mediante una estructura en red de las tareas e hitos del proyecto.**

Las redes de tareas representan las tareas que deben ejecutarse en paralelo y las que deben llevarse a cabo en secuencia debido a una dependencia respecto a la tarea o tareas anteriores.

- Los **nodos rectangulares** representan las **tareas**.
- Los **nodos redondeados** representan los **hitos**

Puedes ver un ejemplo de planificación con red de tareas en la página siguiente:



El diagrama anterior representa la siguiente tabla de dependencias:

Tarea	Duración (días)	Dependencias
T1	8	
T2	15	
T3	15	T1
T4	10	
T5	10	T2, T4
T6	5	T1, T2
T7	20	T1
T8	25	T4
T9	15	T3, T6
T10	15	T5, T7
T11	7	T9
T12	10	T11

Si lo deseas, en lugar de la red de tareas, puedes usar otros modelos de planificación temporal más completos como podrían ser:

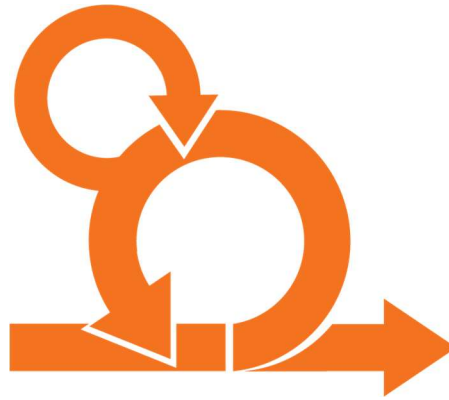
- Diagrama de Pertt.
- Método de Roy.
- CPM.

Sin embargo, los diagramas de Gantt no podrán sustituir a la red de tareas o a los métodos citados anteriormente al no reflejar estas relaciones existentes entre las diferentes actividades. Opcionalmente los diagramas de Gantt pueden incluirse en la documentación del proyecto de forma adicional a uno de los métodos anteriores.

## Especificación de los objetivos de las iteraciones

En el sector de las tecnologías de la información y las comunicaciones (TIC) en el que la velocidad y agilidad al cambio es fundamental se hacen necesarios modelos que nos ayuden a adaptarnos a esos cambios que forman parte de la realidad de este sector.

El desarrollo del proyecto se fundamentará en los principios de las metodologías ágiles. **Las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.**



Las metodologías ágiles se fundamentan en el manifiesto ágil, cuyos 11 principios enumeramos a continuación:

1. Nuestra principal prioridad es satisfacer al cliente a través de la **entrega** temprana y continua del software de valor.
2. Son bienvenidos los **requisitos cambiantes**, aun llegando tarde al desarrollo. Los procesos ágiles se dobligan al cambio como ventaja competitiva para el cliente.
3. Entregar con **frecuencia software que funcione**, en periodos de un par de semanas hasta un par de meses.
4. Las personas del negocio y los desarrolladores deben **trabajar juntos** de forma cotidiana a través del proyecto.
5. Construcción de proyectos en torno a **individuos motivados**.
6. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la **conversación cara a cara**.
7. El software/producto/servicio que **funcione** es la principal medida de progreso.
8. Los procesos ágiles promueven el **desarrollo sostenido**. Los desarrolladores, patrocinadores, y usuarios han de mantener un ritmo constante de forma indefinida.
9. La **atención continua** a la excelencia técnica ensalza la agilidad.
10. **La simplicidad** como arte de maximizar la cantidad de trabajo que se hace, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos que se **auto organizan**.

Elegimos el desarrollo basado en metodologías ágiles debido a las ventajas que nos aportan:

- En primer lugar, las metodologías ágiles **mejoran la satisfacción del cliente** dado que se involucrará y comprometerá a lo largo del proyecto. En cada etapa del desarrollo se informará al cliente sobre los progresos del mismo. De ese modo, el cliente puede sumar su experiencia para optimizar las características del producto final. Se pueden evitar así numerosos malentendidos dado que el cliente poseerá en todo momento una completa visión del estado del producto.
- Asimismo, **mejora la motivación e implicación del equipo de desarrollo**. Pero esta mejora no es casual: las metodologías ágiles permiten a todos los miembros del equipo conocer el estado del proyecto en cualquier momento. Los compromisos son negociados y aceptados por todos los miembros del equipo y las ideas de cualquiera de sus integrantes son tenidas en cuenta.



- Destacar que los procesos ágiles permiten **ahorrar tanto tiempo como costes**. El desarrollo ágil trabaja de un modo más eficiente y rápido que otras metodologías. Además, estos procesos ponen el foco en cumplir estrictamente el presupuesto y los plazos pactados a la hora de definir y planificar el proyecto.
- Se trabaja con **mayor velocidad y eficiencia**. En las metodologías ágiles se trabaja realizando entregas parciales pero funcionales del producto. De ese modo, es posible entregar en el menor intervalo de tiempo posible una versión funcional del producto.
- Gracias a las entregas parciales (centradas en entregar en primer lugar aquellas funcionalidades que en verdad aportan valor) y a la implicación del cliente será posible **eliminar aquellas características innecesarias del producto**.
- Las metodologías ágiles permiten **mejorar la calidad del producto**. La continua interacción entre los desarrolladores y los clientes tienen como objetivo asegurar que el producto final sea exactamente lo que el cliente quiere y necesita. Además, este enfoque permite abrazar la excelencia tecnológica, lo que permite obtener un producto tecnológicamente superior.
- Por otro lado, esta metodología permite **alertar rápidamente tanto de errores como de problemas**. En la etapa de planificación, el equipo ha presentado una hoja de ruta anticipando y dando respuesta a los principales problemas técnicos y a la velocidad en la que se puede trabajar. Con metodologías más tradicionales, los errores no identificados en las primeras fases del proyecto suelen acarrear costes muy altos.
- Y, finalmente, las metodologías ágiles **permiten rentabilizar nuestras inversiones más rápidamente**. Gracias a la realización de entregas tempranas el cliente tendrá rápido acceso a aquellas funcionalidades que en verdad aportan valor acelerando el retorno de la inversión.

Así pues, **deberás establecer una planificación de iteraciones en las que se irán cubriendo paulatinamente los objetivos finales del proyecto, especificando para cada una de ellas:**

- Fecha de inicio y finalización de la iteración.
- Funcionalidades y características a desarrollar en la iteración.
- Planificación temporal de las tareas a desarrollar e hitos a conseguir en base al modelo de red de tareas.

## Prototipado en cada iteración

Para cada una de las iteraciones deberás aportar los cuatro entregables ya explicados para las fases de diseño de la aplicación:

- Sketch
- Wireframe
- Mockup
- Prototipo

En la última iteración de la aplicación podrás observar si existe variación y en que ha consistido esta en relación a la planificación inicial de la aplicación realizada en la fase de la propuesta. **Explica las diferencias y a qué son debidas.**



# DISEÑO DE LA APLICACIÓN

En este apartado reflejarás todo lo relativo al diseño de la aplicación. Incluirás capturas de todas las pantallas y menús de la aplicación final (la correspondiente a la última iteración) y añadirás cuantas explicaciones sean necesarias para su correcta comprensión.

## Esquema funcional

En este apartado incluirás un esquema gráfico, que puede o no ser esquemático, de todas las pantallas de la aplicación conectadas por líneas que parten de los diferentes botones o elementos de menú cuya activación lleva a otras pantallas o paneles emergentes.

## Diseño de las estructuras de datos

En este apartado incluirás las principales estructuras de datos que use tu programa (objetos, matrices, variables relevantes, etc...), así como el esquema entidad-relación de las tablas usadas (si existen), junto con la definición de los tipos de datos usados en los campos de cada tabla.

## Estudio de la interacción del usuario con la aplicación

En este apartado incluirás diagramas de estado de diferentes componentes o secciones de la aplicación que puedan cambiar en base a las acciones que realice el usuario. Por ejemplo, opciones de menús que se habilitan o deshabilitan o controles que aparecen o desaparecen en base a las acciones del usuario, por ejemplo.

## Guía de estilo

La guía de estilo proporciona la **información necesaria para continuar un proyecto digital una vez que la administración es entregada al cliente**. Los estilos son útiles para los diseñadores durante la creación de elementos que conforman la aplicación.

La guía de estilo tiene como objetivo **mantener la congruencia e integridad del diseño de la aplicación**, sin importar quién esté a cargo de su desarrollo y **ayudar a los desarrolladores a comprender cómo debe verse la aplicación**, aun cuando ya ha pasado tiempo desde su implementación inicial.

Para lograr estos objetivos, este entregable debe cumplir con cuatro reglas básicas: ser **intuitivo, sencillo, informativo y compatible con los estándares de CSS**. De esta forma, las interpretaciones arbitrarias quedan excluidas y sólo hay espacio para información objetiva y de fácil uso.

Dentro de las secciones que la componen, se incluye toda la **documentación de diseño y, en algunos casos, instrucciones de usabilidad y legibilidad**. Así, en su creación debemos incluir los siguientes elementos:

1. **Paleta cromática:** Muestra los **colores principales de la aplicación con todas sus posibles variaciones**, ya sean patrones o tonos sólidos. Las referencias deben incluir el código de color exacto e información relevante para su aplicación, como los porcentajes de saturación y luminosidad, cuando estos son necesarios.



2. **Tipografías:** Es un listado de los **tipos de fuentes presentes en el diseño de la aplicación**. Además, explica todas las normas que regirán el uso de estas tipografías en el sitio, incorporando nombres, tamaños, estilos y colores. Es conveniente adjuntar imágenes que ilustren todos estos detalles para evitar confusiones.

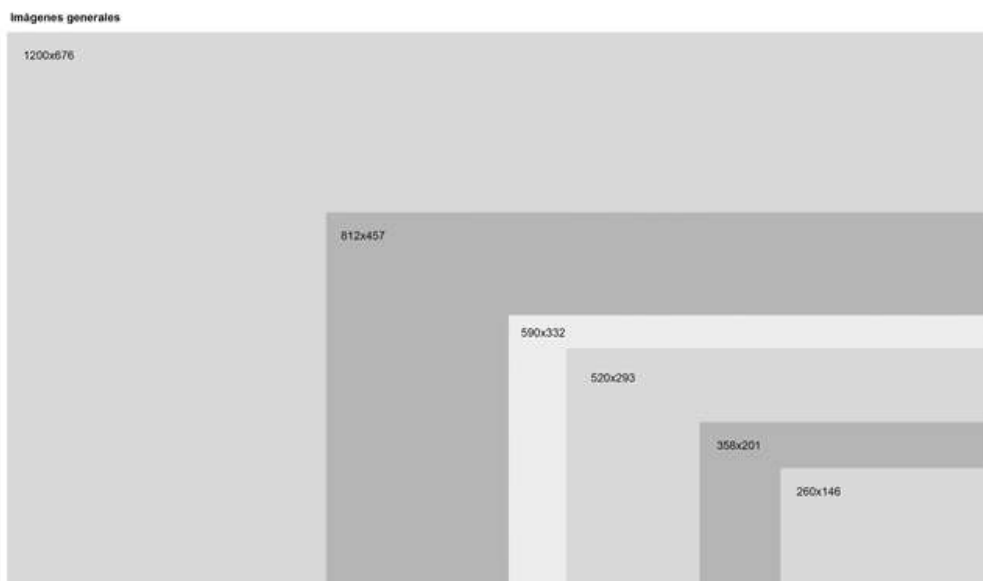


3. **Estilos de botones y/o elementos de navegación:** Define el **uso combinado de las tipografías con la paleta cromática**. Debe mostrar cómo se verán los botones y enlaces, considerando los usos con y sin texto, en hover, selección e inactividad.

También se describen las aplicaciones en los elementos de navegación, mostrando el diseño de las flechas de desplazamiento, breadcrumbs, menús y submenús.



4. **Imágenes:** Esta sección indica todos los **tamaños y proporciones de imágenes (no se consideran las usadas para la estructura de la aplicación) usadas en la aplicación y cómo conviven** dentro del diseño. Además, muestra los efectos o variaciones que puedan tener en sus distintas aplicaciones, especificando si se trata de una galería, foto destacada, uso en el footer u otros.



5. **Otros:** Cada aplicación es diferente, por lo que incluimos un apartado adicional para entregar los **detalles de elementos específicos**. Por ejemplo, explicamos las características de los bloques de autor, calendarios, gráficos, tablas y/o fichas, o cualquier elemento específico cuando estos existan.



Es importante que tu proyecto cuente con una guía de diseño definida, así podrás acceder a la **información necesaria para guiar todas las posibles modificaciones a corto y medio plazo**. Si la utilizas correctamente, el diseño de tu aplicación se mantendrá íntegro y conservará la coherencia al adaptarse a los cambios que necesites realizar en el futuro.

## Vídeo demostrativo

Para un correcto entendimiento del funcionamiento de tu aplicación adjuntarás al proyecto un vídeo con la captura del funcionamiento de la aplicación en cada una de sus funcionalidades. Incluirá locución (puedes usar un programa narrador si lo deseas, siempre que resulte totalmente comprensible) explicando lo que se va mostrando en pantalla.

# CODIFICACIÓN

En este apartado explicarás todo lo relativo a la programación de tu aplicación, aportando un esquema general de los módulos que la componen en el que se ponga de relieve su funcionalidad y relaciones entre los mismos, en caso de existir.

## Lenguajes, herramientas y metodologías empleadas

En este apartado expondrás:

- Cuáles han sido el o los lenguajes de programación utilizados en tu proyecto, justificando su elección.
- Cuáles han sido las herramientas usadas en tu proyecto, sean estas editores de texto, herramientas de codificación o compresión, programas gráficos, etc...
- Cuáles han sido los frameworks, librerías, código de terceros, etc... que has incluido en tu proyecto, aportando una breve descripción de la utilidad de los mismos y la URL desde donde puede accederse a ellos.

## Aspectos destacables de la implementación

Este apartado se incluye con la idea de que puedas explicar aspectos relevantes de tu aplicación que revistan una especial dificultad, que constituyan soluciones interesantes a los retos que has tenido que afrontar, o que consideres importantes por alguna razón que deberás explicar.

## Uso de estándares y normas de accesibilidad

Si tu aplicación cumple con algún tipo de estándar, por ejemplo, supera con éxito una validación del HTML que la forma, o necesita de una versión concreta de ECMAScript para su correcto funcionamiento debes comentarlo en este apartado.

Igualmente, si tu aplicación cumple con alguna de las normas de accesibilidad reconocidas en la actualidad también debes explicar en este apartado todo lo relativo a ese tema. Algunas de las normas de accesibilidad en el ámbito del desarrollo web reconocidas en la actualidad son:

- Norma UNE 139803:2012. Requisitos de Accesibilidad para contenidos en la web.
- Web Content Accessibility Guidelines WCAG, versiones 1.0 y 2.0.
- Estándar ISO/IEC 40500:2012.
- UNE-EN 301 549, “Requisitos de accesibilidad de productos y servicios TIC aplicables a la contratación pública en Europa” – “Accessibility requirements suitable for public procurement of ICT products and services in Europe”.
- WAI-ARIA Accessible Rich Internet Applications.
- Norma UNE 139803:2004. Requisitos de Accesibilidad para contenidos en la web.
- Norma CWA 1554:2006. Especificaciones para el esquema de la evaluación de la conformidad y marca de calidad sobre accesibilidad web.

## DESPLIEGUE DE LA APLICACIÓN

En este apartado explicarás todo lo relativo al despliegue de tu aplicación, describiendo:

- Requisitos para el correcto funcionamiento de la aplicación.
- Configuraciones necesarias, si fuesen necesarias, para el funcionamiento adecuado de la aplicación.
- Creación y configuración de la o las bases de datos necesarias.

## EVALUACIÓN Y PRUEBAS

En este apartado dejarás constancia de las pruebas realizadas a los diferentes módulos de la aplicación y a la aplicación misma en cada una de las iteraciones de su desarrollo.

Igualmente dejarás constancia de los resultados de cada prueba y las correcciones que se llevaron a cabo en la aplicación fruto de los resultados de las pruebas que arrojaron fallos en el funcionamiento de la aplicación.

## MANUAL DE USUARIO

En este apartado redactarás un breve manual de usuario en el que expliques el funcionamiento de la aplicación, así como la forma de ejecutar las funcionalidades de la misma. Este manual estará compuesto por las partes que se describen a continuación.

### **Descripción de la aplicación**

La primera parte del manual de usuario consistirá en la descripción de la aplicación, orientada hacia el usuario, de forma que este pueda hacerse una idea clara de su finalidad y utilidad.

### **Funcionalidades y características**

La segunda parte del manual de usuario consistirá en la explicación, orientada hacia el usuario, de las diferentes funcionalidades y características de la misma, así como la forma de llevarlas a cabo con la aplicación.



## CONCLUSIONES

Este es el apartado final del proyecto de fin de ciclo en el que pondrás de relieve lo aprendido durante su desarrollo, así como las conclusiones más relevantes a las que has llegado tras su finalización.