

# PI 1 - Índice Proyecto

Este es un índice de proyecto de ejemplo para trabajar sobre él

## Índice

### 1. Introducción.

- a. Breve descripción del proyecto y su objetivo (Abstract).  
Debe ser una explicación sencilla del contexto donde desarrollaremos nuestro proyecto.  
Debemos dar respuesta a ¿Qué problemática hay? ¿Cómo se está solucionando?  
Se suele incluir una versión en castellano y otra en inglés
- b. Objetivos del proyecto  
Debemos dar a conocer el principal objetivo del proyecto: ¿Cuál es nuestra propuesta para mejorar o solucionar el problema?
- c. Debemos plantear reflexiones o dudas que se resolverán en el proyecto.

### 2. Estado del arte.

- a. Análisis de trabajos relacionados con el tema del proyecto. Análisis de tecnologías, lenguajes y estrategias o métodos. Sirve para obtener ideas.
- b. Indicar qué novedades no existentes podría aportar tu proyecto (lo vayas a implementar o no), con su justificación.

### 3. Estudio de viabilidad. Método DAFO.

- a. Se analizan, en relación al proyecto, las debilidades a corregir, las amenazas a afrontar, las fortalezas a mantener y las oportunidades a explotar.
- b. Estudio de mercado.
- c. Se analiza la situación actual del mercado, para conocer las tendencias y competencia. ¿Es necesario nuestro producto? ¿Tiene hueco en el mercado? ¿A qué público le puede interesar?
- d. Viabilidad técnica/económica del proyecto
  - i. Se debe evaluar la viabilidad del proyecto a nivel técnico y económico. Para ello, deberemos especificar los recursos a utilizar.
- e. Recursos HW
- f. Recursos SW
- g. Recursos humanos
  - i. Viabilidad temporal
    - a. Se evalúa, a grandes rasgos si el proyecto es viable desde el punto de vista temporal.
  - ii. Planificación temporal o agenda de trabajo.
    - b. Se debe presentar un plan de trabajo (diagrama de Gantt...)

### 4. Análisis de requisitos

- a. Se debe identificar y analizar las necesidades y expectativas del cliente o usuario final en relación al proyecto. Se determinan los objetivos y funcionalidades que el sistema debe cumplir: a nivel frontend, backend y BBDD (requisitos funcionales). Se explicarán a nivel global o general.
- b. Descripción de requisitos.
  - i. Texto explicativo
    - a. Se explicarán los requisitos con sus funcionalidades (página de acceso... formulario que permitirá....)

5. Diagramas de caso de uso de los más relevantes. Realizando un caso de uso general y si es necesario otros diagramas más específicos.

- a. Realizar diagramas o mapas en los que se recogen distintos perfiles de la aplicación atendiendo los usos distintos o condiciones diferentes. Mostrar los más relevantes casos de uso o navegación.

## 6. Diseño

- a. Debemos indicar los elementos de diseño del proyecto. Se deberán realizar antes del proyecto. Son entes dinámicos, podrán variar durante la ejecución del proyecto.
- b. Diseño Conceptual Entidad Relación
- c. Diseño Lógico Relacional o Paso a tablas.
- d. Diseño Físico o Diagrama Mysql
- e. Descripción de las tablas y campos.
- f. Orientación a objetos:
  - i. Diagramas de clases. Descripción de clases y atributos.
  - ii. Diagrama de secuencias. De lo más relevante.
- g. Mockups
- h. Diagrama de actividad. De lo más relevante.
- i. Mapa Web. Gráfico que muestra los enlaces entre páginas.

## 7. Codificación.

- a. Tecnologías elegidas y su justificación (lenguajes, frameworks, bibliotecas, etc.)
- b. Comparación y justificación.
- c. Entorno servidor.
  - i. Descripción general.
  - ii. Seguridad. Evitar inyección en bases de datos.
  - iii. Evitar o capturar errores y warnings.
- d. Entorno cliente.
  - i. Descripción general.
  - ii. Asegurar la funcionalidad en los navegadores más usados (Firefox, Internet Explorer, Safari).
- e. Documentación interna de código.
- f. Código: comentarios, descripción de ficheros, objetos, funciones, como funciona cada parte.
- g. Documentación externa (puede ir en un anexo).
- h. Aquí se puede incluir también Documentación API de la aplicación, manual de despliegue, manual de usuario. Se incluye ejemplo/parte relevante. El resto en anexos.
  - i. Manual del usuario. ¿Accesible desde la web?

## 8. Despliegue

- a. Diagramas de despliegue
  - i. Incluir esquema
- b. Descripción de la instalación o despliegue
  - i. Ficheros de configuración:
    - En php: config.cfg
    - En JAVAEE: web.xml, context.xml.

## 9. Descripción del servidor hosting/cloud utilizado.

- a. NOTA: es recomendable que, en caso de que el alumno quiera que se acceda, para

poder hacer pruebas que indique claramente la url si es en servidor público y el usuario/contraseña de cada uno para poder probar el proyecto.

#### 10. Herramientas de apoyo

#### 11. Control de versiones, depuración, plugins, analizadores de código, linters, transpiladores, entornos de desarrollo, despliegue, juegos de pruebas, etc.

- a. Control de versiones.
- b. Sistemas de integración continua.
- c. Gestión de pruebas
  - i. Explicar como se han realizado pruebas en la aplicación: explicar tipos de pruebas y ejemplos (unitarias, de integración, de usuario, de rendimiento, de escalabilidad...)

#### 12. Conclusiones.

- a. Conclusiones sobre el trabajo realizado
- b. Conclusiones personales
- c. Posibles ampliaciones y mejoras

#### 13. Bibliografía (comentada)

- a. Libros, artículos y apuntes
- b. Direcciones web