

Δ chan

Proyecto final
Desarrollo de Aplicaciones Multiplataforma

Alfredo Rodríguez Gracia

21 de mayo de 2021

última revisión
22 de mayo de 2021

Índice

1. Descripción del proyecto	2
2. Ámbito de implantación	2
3. Recursos de hardware y software	2
4. Temporalización del desarrollo	3
4.1. Diagrama de Gantt	3
4.2. Diagrama PERT	3
5. Descripción de los datos base y resultados	3
6. Relación entre dispositivos y programa o rutinas	3
Referencias	5
Índice de bloques de código	6

1. Descripción del proyecto

Breve descripción del proyecto indicando qué es lo que hace, para qué sirve y si tiene futuros usos con pequeñas modificaciones.

Δ chan (pronunciado como *dichan*) es un proyecto de tablón de imágenes[1], centrado en el anonimato y la libertad de expresión *on-line*, dónde los usuarios pueden subir imágenes y vídeos cortos para iniciar un debate. Este está inspirado en otros tablonos existentes como *4chan* y *2channel*, sitios que, a pesar del enorme auge de las redes sociales actualmente, siguen siendo el refugio de muchos de internautas hoy en día.

Un tablón de imágenes (también conocido por su nombre en inglés: *imageboard*) es un tipo de página web anónima donde la publicación de imágenes y pequeños vídeos cobra una gran importancia. Los primeros tablonos de imágenes fueron creados en Japón a finales de los 90, y se basan en el concepto de los foros de texto. En términos generales ambos comparten la misma estructura, incluyendo la separación de los debates (*threads*) de diferentes temáticas en secciones, llamadas tablonos o *boards*. Sin embargo, los *threads* en los *imageboards* pueden llegar a ser mucho más esporádicos que en los foros convencionales, donde el tiempo de vida de uno puede ser inferior a varias horas. Los tablonos de imágenes más populares en occidente tienden a estar relacionados en su mayoría con la cultura japonesa, como son la temática del *anime* y *manga*. Sin embargo, en Japón son más populares y sus tópicos abarcan una gran variedad de temas.

2. Ámbito de implantación

Deberá describirse el lugar (empresa, organización, sector...) en el que se implantará el proyecto y con qué objetivo, además de indicar a quién va dirigida la aplicación, es decir, identificar quién o quiénes serán los principales usuarios de la misma.

3. Recursos de hardware y software

Se describirán los requisitos mínimos y los requisitos recomendados de hardware, tanto para el desarrollo de la aplicación, como para su instalación y ejecución.

Se describirán las necesidades de software requeridas para el desarrollo de la aplicación.

4. Temporalización del desarrollo

Deben describirse las distintas actividades necesarias para desarrollar el proyecto, asignarles un tiempo a cada una de ellas y construir los dos diagramas completos.

4.1. Diagrama de Gantt

...

4.2. Diagrama PERT

...

5. Descripción de los datos base y resultados

Se describirán el tipo de campo (en caso de java serían: String, char, int, double, long...), que se utilizará para recoger los diferentes datos.

Posibles restricciones y/o estructuras utilizadas (clases). Lo mismo para los datos resultantes de los procesos.

6. Relación entre dispositivos y programa o rutinas

Se identificarán los componentes que comunican el paquete o aplicación software desarrollado con el resto de actores relevantes fuera de la máquina. Es decir, interfaces persona-máquina para entrada y/o salida de datos, interfaces de red u otros medios para comunicación con máquinas remotas, periféricos específicos o componentes concretos de plataformas móviles, etc.

Se identificarán los componentes software (clases, procedimientos) representativos y se vincularán con los anteriormente mencionados a través de texto y/o diagrama(s) que ayuden a comprender el funcionamiento general de la aplicación.

Ejemplo 1: Las funciones de ejemplo

```
1  /*
2   * upperOrNot: Given a character and its predecessor,
3   * it returns that same character converted to upper
4   * or lower case based on the ASCII value of its
5   * predecessor.
```

```

6  */
7  const upperOrNot = (previous, actual) => {
8    if (previous.charCodeAt(0) % 2 !== 0) {
9      return actual.toUpperCase();
10   }
11   return actual;
12 }
13
14 /*
15  * strToNumber: Given a character string returns a
16  * string of numbers, based on the value of the
17  * characters in the ASCII table.
18  */
19 const strToNumber = (str) => {
20   return [...str].map(
21     (char) => char.charCodeAt(0) % 10
22   ).join('');
23 }
24
25 export { upperOrNot, strToNumber };

```

Referencias

- [1] Wikipedia. *Imageboard*. n.d. URL: <https://en.wikipedia.org/wiki/Imageboard>.

Bloques de código

1. Las funciones de ejemplo 3