

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	2
4.1. Diagrama de Gantt	2
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.

Como vemos en el ejemplo 1, las *arrow functions* son preciosas.

Δ chan (pronunciado como *dichan*) es un proyecto de tablón de imágenes, o *imageboard*[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.

La idea está inspirada en otros *imageboards* existentes como *4chan* y *2channel*, sitios muy frecuentados actualmente, a pesar del enorme auge de las redes sociales actualmente.

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

Bloques de código

1. Las funciones de ejemplo 4