



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA INFORMÁTICA

**Asistente para el diseño de la interfaz para control,
seguimiento y sharing de robots en tiempo real**

Manuel López Urbina
Director: Arturo Morgado Estévez

Cádiz, 19 de abril de 2017



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA INFORMÁTICA

RobotUI:

Interfaz de Usuario de propósito general

- Departamento: Ingeniería en Automática, Electrónica, Arquitectura y Redes de Computadores
- Director del proyecto: Arturo Morgado Estévez
- Autor del proyecto: Manuel López Urbina

Cádiz, 19 de abril de 2017

Fdo: Manuel López Urbina

Agradecimientos

Este proyecto significa la culminación de mi carrera, por lo que me gustaría dedicárselo a todas las personas que me han ayudado a conseguir acabarla.

En primer lugar me gustaría agradecerle, de igual modo que en el PFC de la Ingeniería Técnica, a mi familia el apoyarme y ayudarme durante estos años, y el esfuerzo que han hecho para que yo haya podido culminar mi ingeniería.

Mención especial para Natalia Luciano, mi pareja, la cual ha estado siempre apoyándome en mis objetivos y tanta paciencia y comprensión me ha mostrado tras tantos días, meses e incluso años de estudio y dedicación.

Agradecimientos a D. Arturo Morgado por su ayuda y dedicación durante la realización y dirección de este proyecto, así como su carácter amable y servicial que han hecho más ameno el trabajo realizado.

También me gustaría agradecérselo a mis compañeros, con los que tantos ratos inolvidables he pasado, y que tanto me han ayudado.

Por último quiero dedicarle este proyecto a todos los estudiantes de informática, en especial a todos los amantes del fascinante mundo de la robótica, a los que espero que mi trabajo les sea de utilidad.

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyright © 2017 Manuel López Urbina.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Resumen

RobotUI es un proyecto web para el control de dispositivos robóticos y compartirlos con otros usuarios.

Palabras clave: Internet, aplicación web, robótica, robots, interfaz de usuario, streaming de vídeo, control remoto, robot sharing, Sails.js, Node.js, Socket.io, Websockets, compartir.

Índice general

Índice general	I
Índice de figuras	III
1. Introducción	1
1.1. Introducción y antecedentes	1
1.2. Objetivos	3
1.3. Acerca de este documento	4
2. Conceptos básicos	7
2.1. WebSocket	7
2.2. Streaming	7
3. Estado del arte y herramientas utilizadas	9
3.1. Estado del arte de herramientas para monitorizar información	9
3.2. Tecnologías software utilizadas	9
3.2.1. W EX	9
3.2.2. WebStorm	9
3.2.3. Github	10
3.2.4. Git	10
3.2.5. Digital Ocean	10
3.2.6. Node js	11
3.2.7. Sails js	12
3.2.8. Npm	12
3.2.9. SocketIO	12
3.2.10. Bootstrap	13
3.2.11. JQuery	13
3.2.12. Mongo DB	13
3.2.13. Pm2	14
3.3. Tecnologías hardware y materiales utilizados	14
3.3.1. Raspberry Pi Model B	14
3.3.2. Controladora de motores L298N	15
3.3.3. Batería LiPo	16
3.3.4. Tarjeta de expansión con batería de Litio para Raspberry Pi	17
3.3.5. cámara USB de alta definición de 1080p	17

4. Requisitos	19
4.1. Especificación	19
4.1.1. Requisitos funcionales	19
4.1.2. Requisitos no funcionales	20
5. Comunicaciones	21
6. Robot de pruebas	23
7. Organización temporal	25
7.1. Planificación temporal de tareas	28
7.1.1. Hito 1: Planificación y análisis	28
7.1.2. Hito 2: Definición de requisitos	28
7.1.3. Hito 3: Comienzo de desarrollo de la aplicación	29
7.1.4. Hito 4: Desarrollo de la aplicación, módulo componentes	29
7.1.5. Hito 5: Desarrollo de la aplicación, módulo interfaz	29
7.1.6. Hito 6: Desarrollo del módulo de comunicaciones	30
7.1.7. Hito 7: Construcción del vehículo de pruebas	30
7.1.8. Hito 8: Programación del vehículo de pruebas	30
7.1.9. Hito 9: Documentación	30
7.2. Diagrama de Gantt	30
8. Comentarios finales	33
8.1. Presupuesto	33
8.2. Conclusiones	34
8.3. Mejoras futuras	34
Anexos	37
.1. Instalación de Node.js	37
.2. Instalación Sails.js	37
.2.1. Prerrequisitos	37
.2.2. Instalación	38
.3. Instalación de MongoDB	39
.3.1. Prerrequisitos	39
.3.2. Instalación	39
.4. Instalación de WebStorm	41
.5. Robomongo	41
.6. Git	41
.6.1. Prerrequisitos	41
.6.2. Instalación	41
.6.3. Configuración	41
.7. Despliegue	42
Bibliografía	45
GNU Documentation Free License	47

Índice de figuras

1.1. Logo RobotUI ¹	3
1.2. Vehículo utilizado ²	3
3.1. Droplet desplegado en digital ocean	11
3.2. Imagen de una Raspberry Pi 3 Model B	14
3.3. Imagen de la controladora de motores L298n utilizada	16
3.4. Imagen de la batería LiPo utilizada.	16
3.5. Imagen de la tarjeta de expansión con batería de Litio utilizado.	17
3.6. Imagen de la cámara USB utilizada.	17
5.1. Esquema representativo del flujo de conexiones.	22
6.1. Esquema de conexiones del robot de pruebas.	24
7.1. Panel de actividades - Trello	25
7.2. Descomposición de las tareas implicadas en el desarrollo del proyecto (Primera Parte).	27
7.3. Descomposición de las tareas implicadas en el desarrollo del proyecto (Segunda parte).	28
7.4. Diagrama de Gantt 1. Desarrollo del proyecto.	31
7.5. Diagrama de Gantt 2. Desarrollo del proyecto.	32
1. Iniciando Sails ³	38
2. Sails en funcionamiento ⁴	39
3. Utilización del gestor de procesos Pm2 en el entorno de producción. . .	43

Capítulo 1

Introducción

*Lo mejor que podemos hacer por otro
no es sólo compartir con él nuestras riquezas,
sino mostrarle las suyas
Benjamin Disraeli*

1.1. Introducción y antecedentes

La robótica es una rama de la ingeniería, la cual se ocupa del diseño, construcción, operación y uso de robots¹, así como sistemas informáticos para su control, retroalimentación sensorial y procesamiento de información. Entre las diversas disciplinas aplicadas a la robótica podemos encontrar: la mecánica, la electrónica, la informática, la inteligencia artificial, la ingeniería de control y la física, entre otras muchas, de lo cual podemos considerar la robótica como una ciencia multidisciplinar.

En la actualidad, los robots comerciales e industriales son ampliamente utilizados y cada día realizan tareas de forma más exacta o más barata que los humanos. También se les utiliza en trabajos demasiado sucios, peligrosos o tediosos. Los robots son muy utilizados en plantas de fabricación, montaje y embalaje, en transporte, en exploraciones en la Tierra y en el espacio, cirugía, armamento, investigación en laboratorios y en la producción en masa de bienes industriales o de consumo. Otras aplicaciones incluyen la limpieza de residuos tóxicos, minería, búsqueda y rescate de personas y localización de minas terrestres. En definitiva, la robótica está presente en prácticamente cualquier ámbito que podamos imaginar en la actualidad.

Por otra parte, ninguno de los sistemas robóticos actuales podrían ser funcionales sin un software adecuado para su manejo y control, en ocasiones siendo éste tremendamente complejo y específico para garantizar una correcta sincronización entre los diferentes elementos hardware y software implicados con la finalidad de garantizar un correcto

¹Robot: Máquina automática programable capaz de realizar determinadas operaciones de manera autónoma y sustituir a los seres humanos en algunas tareas, en especial las pesadas, repetitivas o peligrosas; puede estar dotada de sensores, que le permiten adaptarse a nuevas situaciones.

funcionamiento del conjunto robótico.

Por estas razones cada vez son más las escuelas que hacen uso de la robótica para que los estudiantes se interesen en la tecnología ya que pueden encontrar un entorno divertido donde aprender, y que ofrece multitud de ventajas:

1. Los niños lo encuentran divertido Hay varios concursos orientados a distintos grupos de edad que pueden canalizar la competencia de una manera positiva. Por ejemplo, se le puede pedir a los niños que construyan un robot y luego hacer competiciones.
2. Es una manera eficaz de enseñarles programación a los estudiantes La programación puede ser muy abstracta. Al tener que controlar un robot físico y ver lo que sale mal, los estudiantes aprenden lo que los robots pueden y no pueden hacer. También aprenden la necesidad de dar instrucciones precisas.
3. Desarrolla habilidades útiles Capacidad de resolución de problemas, trabajo en equipo, capacidad de análisis, y un largo etcétera.

De lo anterior se extrae la necesidad de elaborar un sistema que, además de acercar la robótica a los estudiantes, permita compartir las creaciones con otros usuarios en internet. Todos hemos visto alguna vez vídeos en las redes sociales donde los usuarios nos muestran sus dispositivos en funcionamiento donde, en ocasiones, nos gustaría poder tomar control sobre ellos o visualizar su manejo en tiempo real.

Por tanto el sistema resultante debe cubrir dos necesidades principales, la primera, dotar al usuario de las herramientas necesarias para permitir la configuración de una interfaz de control de sus dispositivos sin necesidad de amplios conocimientos de programación, y la segunda, cubrir la necesidad paralela en la que los usuarios, orgullosos de sus creaciones, dispongan de una manera de compartir sus robots con el resto del mundo de una manera más dinámica. Es decir, en la que otros usuarios, a modo de espectadores, puedan visualizar el control de los dispositivos por parte de su creador, como si de una sesión de vídeo en streaming se tratara. También se dotará de la posibilidad de permitir el control por otros usuarios externos. En la actualidad no existe ningún medio eficaz donde hacer una difusión de la robótica de una manera similar.

Dada la problemática actual presentada, junto con que la programación web y la robótica son temas que causan en mí un especial interés, hicieron que me lanzara a la elaboración de este proyecto que unifica ambos campos anteriormente citados.

Así surgió *RobotUI* y con él un nuevo concepto llamado *RobotSharing*.

Figura 1.1: Logo RobotUI ².

RobotUI (nombre del sistema resultante) será una combinación de un elemento software (aplicación web) y hardware (vehículo de pruebas y demostración) surgido como muestra de la solución obtenida a los citados problemas.

El elemento hardware de este proyecto se compone de un vehículo controlado vía WiFi el cual responde a una serie de señales *comandos* a los que responde realizando determinadas acciones. La interfaz web se configurará de tal manera que permita el control del susodicho vehículo.

Figura 1.2: Vehículo utilizado ³.

1.2. Objetivos

Como hemos visto, se requiere de multitud de conocimientos a la hora de afrontar un proyecto robótico con ciertas garantías. Este proyecto trata, al menos, de reducir, o facilitar, el área relacionada con la informática, más concretamente con la programación. En la que multitud de personas ven en la programación un impedimento a la hora de comenzar a desarrollar sus ideas. Por otro lado, existe la imperiosa necesidad de que la comunidad quiera mostrar sus creaciones al resto del mundo, compartir experiencias, problemas opiniones, etc, de una forma directa y no mediante la grabación de vídeos del funcionamiento de los proyectos robóticos en cuestión, ya que no disponen de una herramienta adecuada para ello. En definitiva, existe la necesidad de que otros usuarios puedan participar de manera más activa, ya sea visualizando el control por su creador o permitir que otros usuarios tomen el control de esos proyectos en tiempo real. Por tanto este proyecto busca facilitar la árdua labor de programación de los proyectos robóticos

²Logotipo RobotUI.

³Vehículo utilizado.

junto con la posibilidad de compartir las creaciones realizadas con otros usuarios.

El sistema a desarrollar, por tanto, dispondrá de dos modos de funcionamiento, el primero de ellos proporciona las herramientas para la configuración de una interfaz a gusto del usuario, en la cual, una vez configurada, el usuario podrá controlar a su antojo el dispositivo. En el segundo modo de funcionamiento, la aplicación permitirá que otros usuarios puedan visitar la interfaz anteriormente configurada y actuar como espectadores en el control del robot por el usuario propietario del mismo. En definitiva, se proporcionará un sistema de control y difusión en uno solo.

1.3. Acerca de este documento

El documento se ha sido elaborado en un lenguaje claro y sencillo para permitir que un estudiante universitario de Ingeniería Informática pueda comprender los contenidos sin apenas dificultad añadida.

Este documento se organiza en los siguientes capítulos:

- En el capítulo 1, Introducción, se comentan las razones que han motivado la creación de este proyecto, así como el propósito del mismo.
- En el capítulo 2, Conceptos básicos, se incluyen definiciones de aquellos conceptos considerados de interés para la correcta comprensión del contenido de la presente memoria.
- En el capítulo 3, Estado del arte y herramientas utilizadas, se realiza una descripción de las diferentes elementos hardware y software empleados durante el desarrollo del proyecto y necesarios para la utilización del mismo. Así como una breve descripción del conocimiento acumulado y tecnologías existentes hasta la fecha.
- En el capítulo 4, Especificación y análisis de requisitos, se realiza una descripción de las diferentes elementos hardware y software empleados durante el desarrollo del proyecto y necesarios para la utilización del mismo. Así como una breve descripción del conocimiento acumulado y tecnologías existentes hasta la fecha.
- En el capítulo 7, Organización temporal, se recoge todo lo que concierne a la distribución y duración de cada una de las tareas llevadas a cabo durante el desarrollo del proyecto que el presente documento describe.
- En el capítulo ??, Configuración y montaje de los dispositivos hardware, se explica el proceso seguido para la correcta integración de los dispositivos hardware empleados describiendo la interconexión entre ellos así como su configuración.

- En el capítulo ??, Desarrollo software, se realiza un análisis sobre la metodología empleada para el desarrollo software, describiendo los modelos de ciclo de vida utilizados, la descripción de los requisitos funcionales junto con el diagrama de casos de uso.
- En el capítulo ??, Software de reconocimiento, se hace una descripción explicando los diferentes aspectos y elementos de cada uno de los prototipos desarrollados junto con los problemas encontrados y soluciones adoptadas.
- En el capítulo ??, Software de control, se describe cómo se ha llevado a cabo la comunicación ordenador-vehículo a nivel software.
- En el capítulo ??, Interfaz gráfica, se recogen aquellos aspectos técnicos de interés referentes a la elaboración de la interfaz gráfica.
- En el capítulo ??, Guía de usuario, se describen los diferentes aspectos necesarios para la correcta utilización del conjunto software y hardware de los que se compone el presente proyecto.
- En el capítulo 8, Conclusiones, se hace mención de las conclusiones obtenidas tras la realización del proyecto además de las posibles mejoras aplicables.
- En el capítulo Anexos 8.3, aparecen los manuales de instalación del software que ha sido necesario para la realización del proyecto.

Capítulo 2

Conceptos básicos

2.1. WebSocket

WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servidores web, pero puede ser utilizada por cualquier aplicación cliente/servidor.

La API de WebSocket está siendo normalizada por el W3C, mientras que el protocolo WebSocket ya fue normalizado por la IETF¹ como el RFC 6455.

Debido a que las conexiones TCP comunes sobre puertos diferentes al 80 son habitualmente bloqueadas por los administradores de redes, el uso de esta tecnología proporcionaría una solución a este tipo de limitaciones proveyendo una funcionalidad similar a la apertura de varias conexiones en distintos puertos, pero multiplexando diferentes servicios WebSocket sobre un único puerto TCP a costa de una pequeña sobrecarga del protocolo.

2.2. Streaming

La retransmisión (en inglés streaming, también denominado transmisión) es la distribución digital de contenido multimedia a través de una red de computadoras, de manera que el usuario utiliza el producto a la vez que se es descargado. La palabra retransmisión se refiere a una corriente continua que fluye sin interrupción, habitualmente audio o vídeo, aplicándose la difusión de vídeo en el presente proyecto.

Este tipo de tecnología funciona mediante un búfer de datos que va almacenando el flujo de descarga en la estación del usuario para inmediatamente mostrarle el material

¹Internet Engineering Task Force (IETF) (en español, Grupo de Trabajo de Ingeniería de Internet) es una organización internacional abierta de normalización, que tiene como objetivos el contribuir a la ingeniería de Internet, actuando en diversas áreas, como transporte, encaminamiento, seguridad. Se creó en los Estados Unidos, en 1986. Es mundialmente conocido porque se trata de la entidad que regula las propuestas y los estándares de Internet, conocidos como RFC.

descargado. Esto se contrapone al mecanismo de descarga de archivos, que requiere que el usuario descargue los archivos por completo para poder acceder al contenido.

La retransmisión requiere de una conexión por lo menos de igual ancho de banda que la tasa de transmisión del servicio. La retransmisión de vídeo por Internet se popularizó a fines de la década de 2000, cuando la contratación del suficiente ancho de banda para utilizar estos servicios en el hogar se hizo lo suficientemente barato.

Capítulo 3

Estado del arte y herramientas utilizadas

3.1. Estado del arte de herramientas para monitorizar información

3.2. Tecnologías software utilizadas

A continuación se detallan las diferentes tecnologías/bibliotecas/lenguajes que se han empleado para la elaboración del proyecto y por qué se han escogido por encima de otras posibles soluciones.

3.2.1. \LaTeX

Web: <https://www.latex-project.org/>

\LaTeX es un lenguaje de marcado que sirve para la redacción de documentos científicos o técnicos. Con esta herramienta o lenguaje se ha desarrollado la memoria actual del proyecto de final de carrera.

3.2.2. WebStorm



Web: <https://www.jetbrains.com/webstorm/>

WebStorm es un IDE de JavaScript ligero pero potente, perfectamente equipado para el desarrollo del lado del cliente y el desarrollo del servidor con Node.js. Permite la inte-

gración con frameworks de desarrollo como Sails js. Para el desarrollo de la aplicación se optó por este IDE.

3.2.3. Github



Web: <https://about.github.com/>

Repositorio: <https://github.com/lopi87/SAILS-RobotUI>

GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Utiliza el framework Ruby on Rails por GitHub, Inc. (anteriormente conocida como Logical Awesome). Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

3.2.4. Git



Web: <https://git-scm.com/>

Git es un sistema open-source de control de versiones diseñado para manejar integralmente las fases de desarrollo de proyectos, simples y complejos, con velocidad y eficiencia.

3.2.5. Digital Ocean



Web: <https://www.digitalocean.com/>

Servidor web para alojar proyectos en la nube. La ventaja de este servicio de VPS es que te permite desplegar máquinas de cualquier tipo (siempre que sean software libre) de una manera muy fácil y rápida. Además tiene un punto fuerte y es que la información se almacena en discos SSD, con lo que el procesamiento se ve muy mejorado a la hora

de computar (en este caso trabajo con web sockets y transmisión de datos).

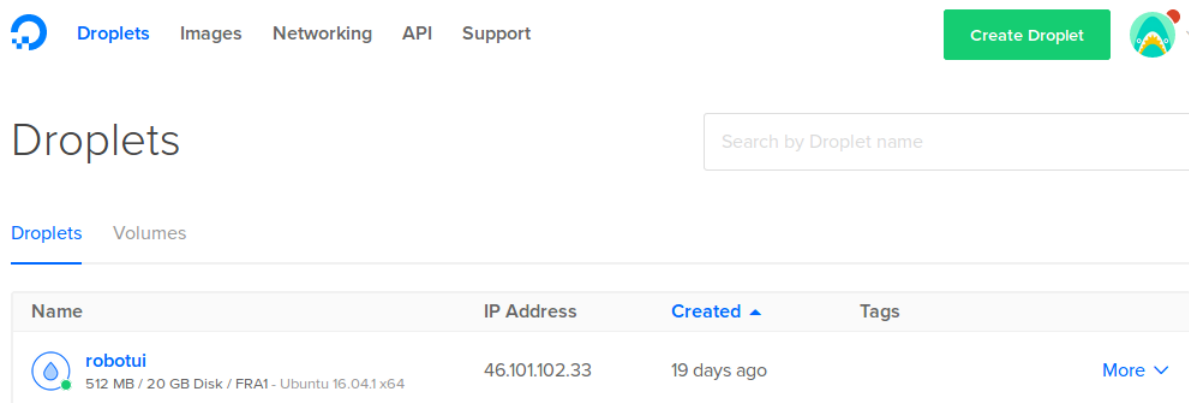


Figura 3.1: Droplet desplegado en digital ocean

3.2.6. Node js



Web: <https://nodejs.org/es/>

Node.js es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. Incorpora un sistema de gestión de paquetes llamado, npm, es el ecosistema mas grande de librerías de código abierto en el mundo.

Node.js tiene una arquitectura basada en eventos capaz de E/S asíncronos. Estas opciones de diseño apuntan a optimizar el rendimiento y la escalabilidad en aplicaciones Web con muchas operaciones de entrada/salida, así como para aplicaciones Web en tiempo real (por ejemplo, programas de comunicación en tiempo real y juegos de navegador), lo que lo hacen ideal para este proyecto.

3.2.7. Sails js



Web: <http://sailsjs.com/>

Sails.js es un framework web que facilita la creación de aplicaciones personalizadas Node.js de nivel empresarial. Está diseñado para parecerse a la arquitectura MVC de frameworks como Ruby on Rails, pero con soporte para el estilo de desarrollo de aplicaciones web más moderno y orientado a datos.

Utiliza Express para funciones como la gestión de peticiones HTTP y websockets. Su envoltorio intuitivo de websockets lo hace especialmente bueno para construir características en tiempo real como por ejemplo un Chat, por estas razones se ha considerado como el framework más adecuado hasta la fecha para la elaboración de este proyecto.

3.2.8. Npm



Web: <https://www.npmjs.com/>

npm es el manejador de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript. Utilizado para la descarga de las librerías incorporadas al proyecto.

3.2.9. SocketIO



Web: <https://socket.io/>

Socket.io es una librería que nos permite manejar eventos en tiempo real mediante una conexión TCP y todo ello en JavaScript desde un cliente. Es realmente potente y podemos hacer todo tipo de aplicaciones en tiempo real.

Socket.IO es una biblioteca de JavaScript para aplicaciones web en tiempo real. Permite la comunicación bidireccional en tiempo real entre clientes web y servidores. Consta de dos partes: una biblioteca del lado del cliente que se ejecuta en el navegador y una

biblioteca del lado del servidor para Node.js. Ambos componentes tienen una API casi idéntica. Al igual que Node.js, es impulsado por eventos.

Socket.IO puede usarse simplemente como un wrapper para WebSocket aunque proporciona muchas más funciones, incluyendo la transmisión a múltiples sockets, almacenamiento de datos asociados a cada cliente y E/S asíncronas.

3.2.10. Bootstrap



Web: <http://getbootstrap.com/>

Bootstrap es un framework o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales. Se ha utilizado en el presente proyecto para la maquetación de la aplicación.

3.2.11. JQuery



Web: <https://jquery.com/>

jQuery es una biblioteca de JavaScript rápida, pequeña y característica. Hace que las cosas como HTML documento transversal y manipulación, manejo de eventos, animación, y Ajax mucho más simple con una fácil de usar API que funciona a través de una multitud de navegadores. Con una combinación de versatilidad y extensibilidad, jQuery ha cambiado la forma en que millones de personas escriben JavaScript.

3.2.12. Mongo DB



Web: <https://www.mongodb.com/es>

3.3. TECNOLOGÍAS HARDWARE Y MATERIALES UTILIZADOS

MongoDB (de la palabra en inglés “humongous” que significa enorme) es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto.

MongoDB forma parte de la nueva familia de sistemas de base de datos NoSQL. En lugar de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en documentos similares a JSON con un esquema dinámico (MongoDB utiliza una especificación llamada BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

3.2.13. Pm2



Web: <http://pm2.keymetrics.io/>

PM2 es un gestor de procesos de producción para aplicaciones Node.js con un equilibrador de carga incorporado. Le permite mantener las aplicaciones vivas para siempre, recargarlas sin tiempo de inactividad y facilitar las tareas comunes del administrador del sistema.

3.3. Tecnologías hardware y materiales utilizados

3.3.1. Raspberry Pi Model B



Figura 3.2: Imagen de una Raspberry Pi 3 Model B

Web: <https://www.raspberrypi.org>

La Raspberry Pi 3 es la tercera generación de Raspberry Pi. Sus especificaciones son las siguientes:

- Una CPU ARMv8 quad-core de 64 bits de 64 bits y 1.2 GHz
- LAN inalámbrica 802.11n
- Bluetooth 4.1
- Bluetooth baja energía (BLE)
- 1 GB de RAM
- 4 puertos USB
- 40 conexiones GPIO
- Puerto HDMI
- Puerto Ethernet
- Conector de audio combinado de 3,5 mm y vídeo compuesto
- Interfaz de la cámara (CSI)
- Interfaz de pantalla (DSI)
- Ranura para tarjeta Micro SD
- VideoCore IV núcleo de gráficos 3D

3.3.2. Controladora de motores L298N

El módulo controlador de motores L298N H-bridge nos permite controlar la velocidad y la dirección de dos motores de corriente continua o un motor paso a paso de una forma muy sencilla, gracias a los 2 los dos H-bridge que dispone.

básicamente un puente-H o H-bridge es un componente formado por 4 transistores que nos permite invertir el sentido de la corriente, y de esta forma podemos invertir el sentido de giro del motor.

El rango de tensiones en el que trabaja este módulo va desde 3V hasta 35V, y una intensidad de hasta 2A. A la hora de alimentarlo hay que tener en cuenta que la electrónica del módulo consume unos 3V, así que los motores reciben 3V menos que la tensión con la que alimentemos el módulo.

Además el L298N incluye un regulador de tensión que nos permite obtener del módulo una tensión de 5V, perfecta para alimentar nuestro Arduino. Eso sí, este regulador sólo funciona si alimentamos el módulo con una tensión máxima de 12V.

Es un módulo que se utiliza mucho en proyectos de robótica, por su facilidad de uso y su reducido precio.

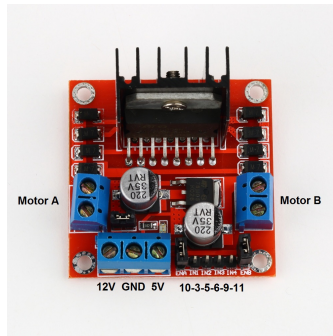


Figura 3.3: Imagen de la controladora de motores L298n utilizada

3.3.3. Batería LiPo

Para alimentar el conjunto se ha empleado una batería LiPo de 1000mAh a 3,7V. Las batería de polímero de iones de litio, son pilas recargables (células de secundaria), compuestas generalmente de varias células secundarias idénticas en paralelo para aumentar la capacidad de la corriente de descarga. Siendo ideales para este tipo de usos.



Figura 3.4: Imagen de la batería LiPo utilizada.

3.3.4. Tarjeta de expansión con batería de Litio para Raspberry Pi

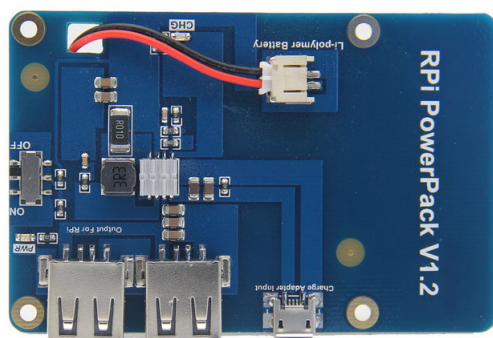


Figura 3.5: Imagen de la tarjeta de expansión con batería de Litio utilizado.

3.3.5. cámara USB de alta definición de 1080p



Figura 3.6: Imagen de la cámara USB utilizada.

Capítulo 4

Especificación y análisis de requisitos

Para el caso que nos concierne en el proyecto, dentro del marco de investigación que define la totalidad de la infraestructura, la funcionalidad principal del mismo será:

- Definir los pasos para dar de alta un dispositivo robótico en el sistema.
- Una vez configurado el dispositivo, configurar la interfaz de control con las acciones de control específicas.
- Realizar un sistema de monitorización y visualización para los usuarios espectadores en tiempo real.
- Sistema de gestión de base de datos en donde se encuentren los datos de la aplicación recogidos.
- Panel de administración donde visualizar la información de los usuarios conectados y dispositivos en uso en tiempo real.

4.1. Especificación de los requisitos

En esta etapa del modelado de requisitos se captura el propósito general del sistema:

- Se analiza qué debe hacer el sistema.
- Se obtiene una versión contextualizada del sistema.
- Identifica y delimita el sistema.
- Se determinan las características, cualidades y restricciones que debe satisfacer el sistema.

4.1.1. Requisitos funcionales

Los requisitos funcionales que se han obtenido en el sistema son los siguientes:

- Ser una herramienta multiplataforma y que permita a cualquier usuario definir sus propias interfaces para el control de robots.

- Dotar de funcionalidad gráfica que permita en tiempo real con mecanismos visuales (en web) visualizar el control de dispositivos robóticos por parte de otros usuarios, modo espectador de la aplicación.
- Proporcionar un sistema de streaming de vídeo para la difusión de imágenes a los usuarios espectadores procedentes de los robots dispongan de cámara.
- Implementar un panel de administración para la visualización de usuarios y dispositivos conectados en tiempo real.

4.1.2. Requisitos no funcionales

Los requisitos no funcionales son aquellos que describen cualidades o restricciones del sistema que no se relacionan de forma directa con el comportamiento funcional del mismo. A continuación se especifican los más importantes del sistema:

- No requiere un conocimiento específico del sistema una vez puesto en funcionamiento.
- La aplicación tendrá manual de uso.
- La base de datos estará implementada en un lenguaje objeto no relacional como MongoDB.
- La aplicación estará realizada en el lenguaje de programación Python.
- La interfaz debe reflejar claramente la distinción entre las distintas partes del sistema.
- El sistema se desplegará sobre una versión GNU Linux Debian 8 Jessie.
- El código fuente de la aplicación seguirá un estilo uniforme y normalizado para todos los módulos del mismo.

Capítulo 5

Comunicaciones

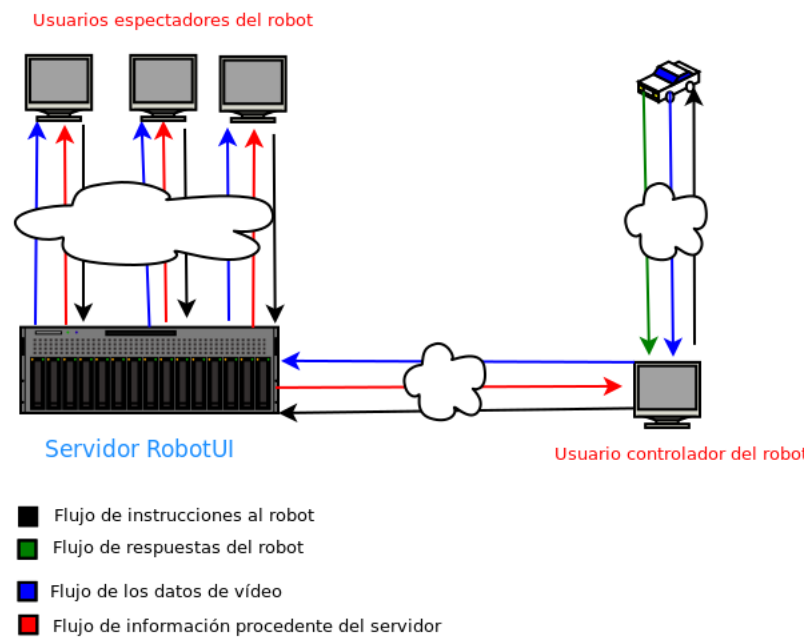


Figura 5.1: Esquema representativo del flujo de conexiones.

Capítulo 6

Robot de pruebas

Capítulo 7

Organización temporal

La planificación general del proyecto siguiendo un modelo SCRUM, empleando para ello el panel de tareas Trello; un gestor de proyectos que permite aplicar una metodología de desarrollo ágil.

El panel se encuentra accesible en el siguiente enlace: <https://trello.com/b/SpIbFI7k/robotui>.

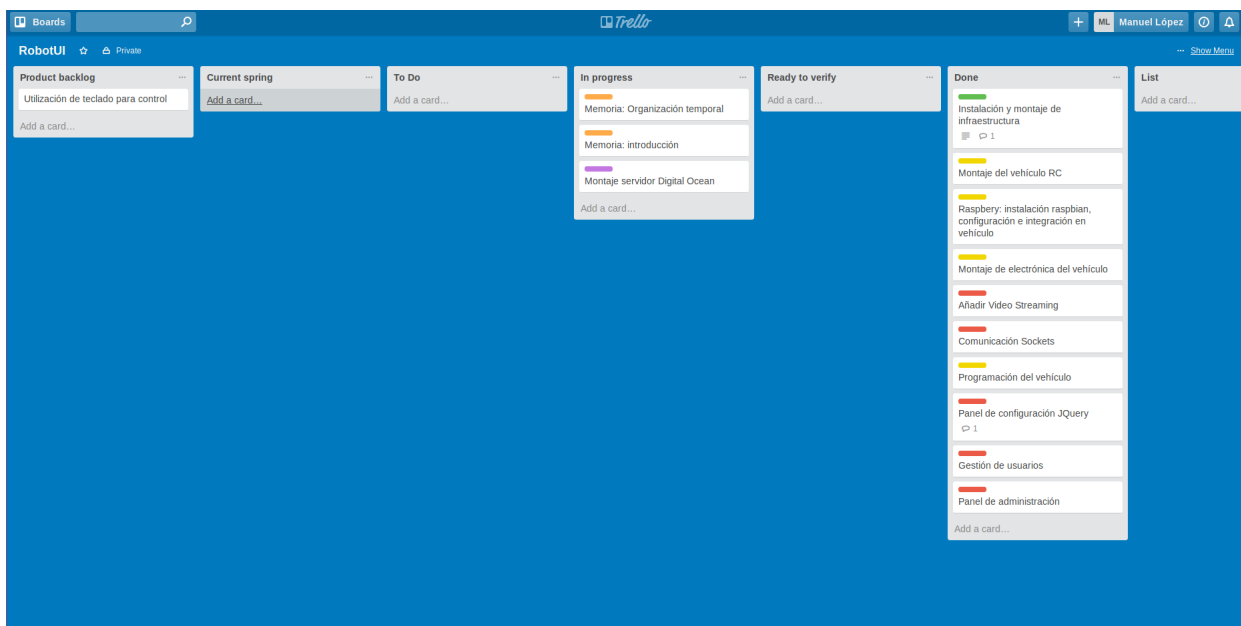


Figura 7.1: Panel de actividades - Trello

Cabe destacar que para el desarrollo de RobotUI ha sido necesario emplear varias herramientas, utilidades y bibliotecas. Algunas de ellas ya habían sido utilizadas en ciertas ocasiones, bien sea en el ámbito estudiantil o profesional. Sin embargo, otras han requerido un periodo de formación previo, en el que se han adquirido los conocimientos necesarios para poder desarrollar el presente proyecto.

La mayor parte del proceso de investigación fue dedicado al estudio de las diferentes tecnologías existentes para la programación web en tiempo real. Todo ello ha implicado

un esfuerzo bastante considerable en el uso, aprendizaje e investigación de las diferentes tecnologías existentes y comprobar su potencial.

Una vez determinadas las diferentes herramientas a utilizar se comenzó con la implementación de la aplicación, siendo seleccionada como herramienta principal el framework Sails.js. Un framework MVC en tiempo real para Node.js, el cual está muy enfocado al propósito de este proyecto.

Finalmente, tras la necesidad de probar la aplicación en un entorno real, se optó por elaborar un robot de pruebas, un pequeño vehículo elaborado con una Raspberry Pi con la finalidad de probar, testear y hacer demostraciones de la aplicación.

La figura 7.2 y 7.3 muestran una visión de las diferentes tareas desarrolladas para la elaboración del proyecto junto con la descomposición de cada una de ellas:

Los puntos más importantes del proyecto se han dividido en hitos, así como entregas que se definieron en cada reunión que se realizaba con el director del proyecto. También se ha definido una planificación temporal del desarrollo del proyecto mediante un diagrama de Gantt con la duración de las tareas recogidas en el panel de actividades de Trello.

WBS	Nombre	Trabajo
1	▼ RobotUI	135d 3h
1.1	Conocimiento del proyecto	22d
1.2	Planificación y estudio del proyecto	5d
1.3	▼ Análisis de herramientas existentes	17d
1.3.1	Estudio de Node.js	5d
1.3.2	Estudio de Sails.js	5d
1.3.3	Estudio de Socket.io	2d
1.3.4	Códigos y pruebas	3d
1.3.5	Estudio de MongoDB	2d
1.4	▼ Definición de requisitos	5d 6h
1.4.1	Arquiterura BBDD análisis	1d
1.4.2	Diseño de diagrama UML	2d
1.4.3	Requisitos funcionales	1d
1.4.4	Requisitos no funcionales	6h 45min
1.4.5	Reunión de planificación con director del proyecto	1d
1.5	▼ Desarrollo aplicación	30d 4h
1.5.1	Sistema de autenticación - registro	2d
1.5.2	Alta de dispositivos robóticos	2d
1.5.3	Internacionalización	1d
1.5.4	Módulo de mensajes entre usuarios	3d
1.5.5	Implementación de políticas de permisos	4d
1.5.6	Reunión de seguimiento con director del proyecto	3d 3h
1.5.7	▼ Elementos de la interfaz	5d 5h
1.5.7.1	Acciones	1d 1h
1.5.7.2	Sliders	1d
1.5.7.3	Labels	1d 7h
1.5.7.4	Vídeo	1d 4h
1.5.8	▼ Personalización de la interfaz	9d 4h
1.5.8.1	Personalización de acciones	1d 4h
1.5.8.2	Personalización de sliders	2d
1.5.8.3	Personalización de labels	2d
1.5.8.4	Edición de la interfaz	4d
1.6	▼ Módulo de comunicaciones	18d
1.6.1	▼ Conexión cliente - robot	4d 7h
1.6.1.1	Envío de órdenes al robot	2d
1.6.1.2	Captura de vídeo del robot	2d 7h
1.6.2	▼ Conexión cliente - servidor	3d
1.6.2.1	Envío de vídeo capturado al servidor	1d 4h
1.6.2.2	Envío de órdenes lanzadas al servidor	1d 4h
1.6.3	Desarrollo de tests funcionales	8d
1.6.4	Frontend - detalles de estilos	2d

Figura 7.2: Descomposición de las tareas implicadas en el desarrollo del proyecto (Primera Parte).

1.7	▼ Conexión servidor - cliente	4d
1.7.1	Difusión de vídeo a espectadores	2d
1.7.2	Difusión de comandos a espectadores	2d
1.8	▼ Montaje del vehículo	3d
1.8.1	Búsqueda de elementos hardware	2d
1.8.2	Montaje y conexiones	1d
1.9	▼ Programación del vehículo	4d 7h
1.9.1	Gpio	2d
1.9.2	Video streaming	1d 3h
1.9.3	Fase de pruebas	1d 4h
1.10	Despliegue de la aplicación en producción	1d
1.11	▼ Documentación	23d
1.11.1	Memoria	19d
1.11.2	Resumen	2d
1.11.3	Presentación	2d
1.12	Reunión de seguimiento con el director del proyecto	1d

Figura 7.3: Descomposición de las tareas implicadas en el desarrollo del proyecto (Segunda parte).

7.1. Planificación temporal de tareas

A continuación se definirán los diferentes hitos que componen el diagrama de Gantt divididos en las diferentes subtareas principales de cada uno de ellos:

7.1.1. Hito 1: Planificación y análisis

En esta primera etapa de desarrollo del proyecto final de carrera se realizaron los estudios previos necesarios para abordar cualquier proyecto de cierta envergadura.

Este hito se descompone en las siguientes tareas principales:

1. Planificación y estudio del proyecto. En esta fase se centró en la elaboración de un documento, a modo borrador, con la idea a desarrollar, objetivos del proyecto y su alcance.
2. Análisis de herramientas existentes, de las tecnologías a implementar, la arquitectura del sistema, las tecnologías de BD, visualización, para la selección de las herramientas más adecuadas para afrontar el desarrollo con garantías y no sea necesaria una “vuelta atrás” por necesidad imperiosa de cambio de tecnología. En definitiva se buscaba una herramienta libre, con un respaldo de una comunidad importante y que resuelva la problemática o necesidad de trabajar con eventos en tiempo real.

7.1.2. Hito 2: Definición de requisitos

Este segundo hito queda dividido en las siguientes etapas:

1. Elaboración de un documento formal con la propuesta de proyecto definiendo sus objetivos y alcance, para la aprobación por parte del director del proyecto.
2. Se definen las clases del sistema, el modelo de la base de datos y la definición de requisitos funcionales y no funcionales.

7.1.3. Hito 3: Comienzo de desarrollo de la aplicación

En este tercer hito, uno de los de mayor magnitud, se comienza con el desarrollo de la aplicación, el cual queda dividido en las siguientes módulos.

1. Implementación del módulo *Usuario* con las funciones de registro, autenticación, configuraciones de idioma, entre otras.
2. Implementación del módulo central de la aplicación *Robot*
3. Implementación del módulo de mensajes.
4. Implementación del módulo de políticas de permisos.

7.1.4. Hito 4: Desarrollo de la aplicación, módulo componentes

En este hito, se trabaja en el desarrollo de los diferentes elementos que compondrán la interfaz de control. Tanto su parte de configuración y personalización. Cada elemento integrante de la interfaz lo denominamos componentes.

En definitiva, el hito queda dividido en el desarrollo de los siguientes componentes:

1. *Acciones*
2. *Sliders*
3. *Labels*
4. *Vídeo*

7.1.5. Hito 5: Desarrollo de la aplicación, módulo interfaz

En este hito, se realiza la elaboración de la interfaz de control. Tanto su parte de configuración y personalización como la de visualización para su control. Este hito quedó dividido en las siguientes etapas:

1. Implementación de la funcionalidad de configuración
2. Implementación de la funcionalidad de control

7.1.6. Hito 6: Desarrollo del módulo de comunicaciones

El presente hito, clasificado como crítico debido a su importancia. Debía realizar la integración de todos los módulos anteriores y dotarlos de la funcionalidad principal para la que han sido diseñados. Estar interconectados entre sí además de la elaboración de los test funcionales.

1. Implementación de la conexión cliente - robot.
2. Implementación de la conexión cliente - servidor.
3. Desarrollo de test funcionales.

7.1.7. Hito 7: Construcción del vehículo de pruebas

Se realiza la construcción y montaje e instalación software del vehículo de pruebas y comprobación de las diferentes conexiones.

7.1.8. Hito 8: Programación del vehículo de pruebas

Se realiza la programación del vehículo y se realizan pruebas de todo el conjunto junto con las correcciones necesarias.

7.1.9. Hito 9: Documentación

Se finaliza la memoria para la revisión por parte del director del proyecto y su posterior impresión. Se prepara la presentación para la defensa ante tribunal.

7.2. Diagrama de Gantt

A continuación se muestra el diagrama de Gantt donde quedan reflejados los diferentes hitos descritos en el punto anterior.

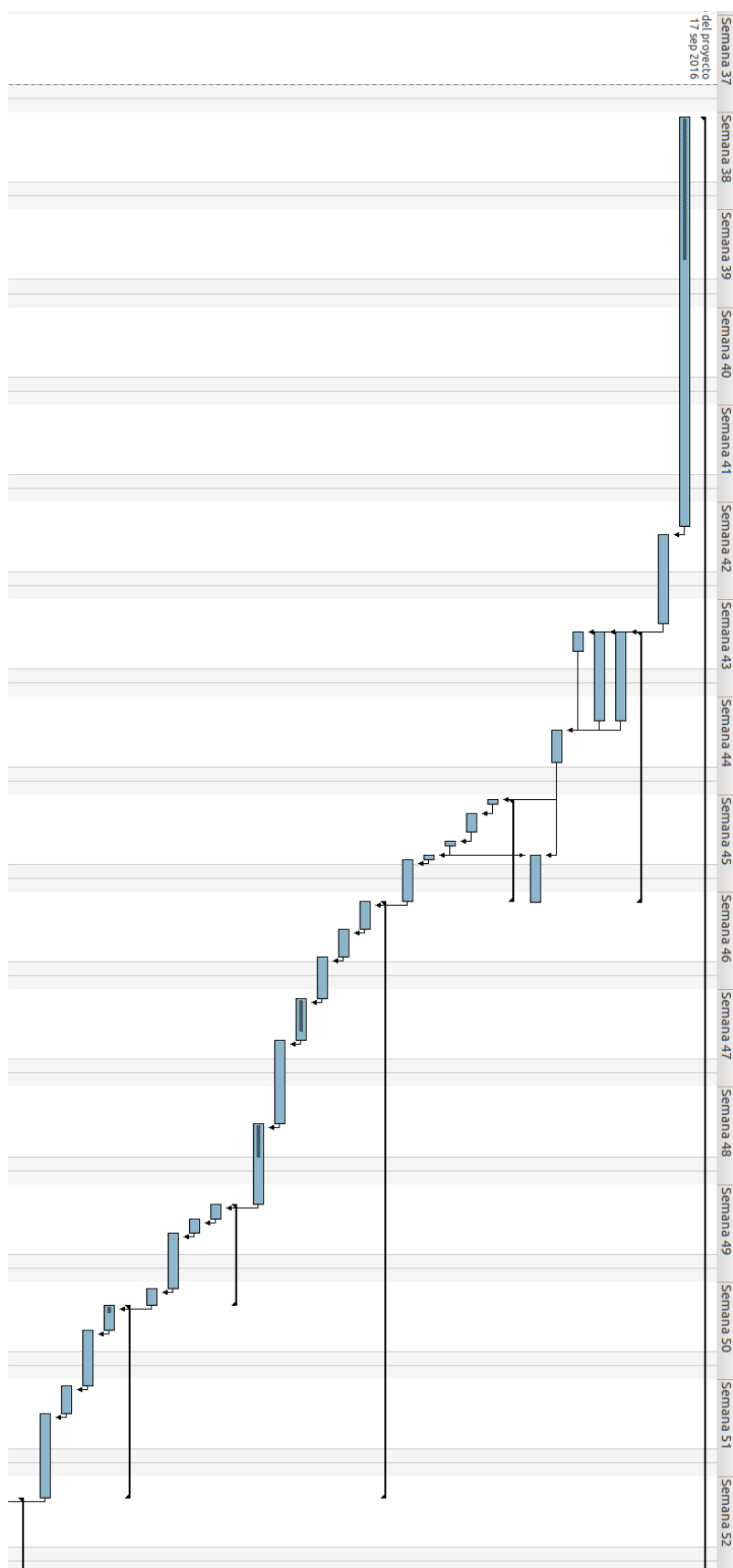


Figura 7.4: Diagrama de Gantt 1. Desarrollo del proyecto.

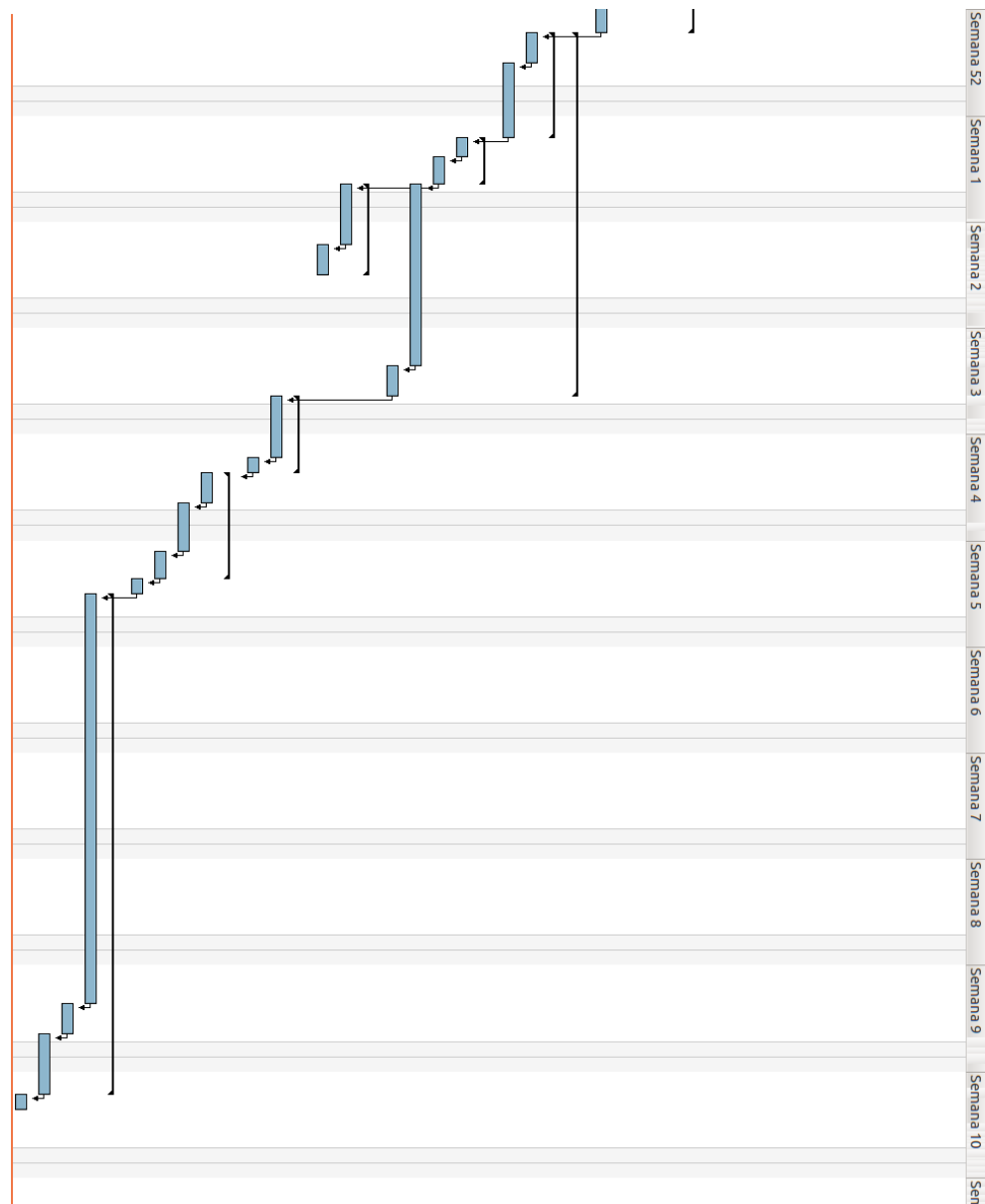


Figura 7.5: Diagrama de Gantt 2. Desarrollo del proyecto.

Capítulo 8

Comentarios finales

8.1. Presupuesto

Descripción	Unidades	Precio €	Total €
Raspberry Pi 3 Modelo B	1	38,7	38,7
Lipo batería (3.7v, 600mAh Lipo)	1	13,99	13,99
Indicador Tester de baterías Lipo	1	8,90	8,90
Cargador baterías Lipo	1	21,90	21,90
Vehículo Radiocontrol	1	54	95
Droplet Digital Ocean	2	5/mes	10
Horas de programación	350	50/hora	17500

Total bruto:	17688,49 €
I.V.A. %:	21 %
Total presupuesto:	21403,07 €

8.2. Conclusiones

La elaboración de este proyecto ha resultado muy gratificante a nivel personal. Uno de los motivos principales ha sido la necesidad de trabajar en numerosas áreas de conocimiento entre las que encontramos, por un lado la programación web, haciendo uso del framework Sails js, junto con el empleo de una base de datos no relacional. Todo ello combinado con la robótica. Algunas de las plataformas mencionadas eran desconocidas al inicio del desarrollo de proyecto y han sido adquiridas tras una amplia labor de investigación.

Entre los elementos desarrollados se destaca:

- La elaboración de un vehículo de pruebas haciendo uso de una Raspberry Pi+
- Aprendizaje a la utilización del framework Sails.js
- Aprendizaje al trabajo con eventos en tiempo real mediante el empleo de Web-Sockets, tecnología nunca utilizada por mí hasta la fecha
- Empleo de una base de datos no relacional como Mongo DB.
- Transmisión de gran cantidad de datos entre cliente servidor y servidor cliente. Streaming de vídeo y emisión de comandos entre otros datos.

Pienso que el resultado final del proyecto es ideal para aquellas personas aficionadas a la robótica y programación proporcionando una herramienta sea utilizable por la gran comunidad poseedora del de cualquier proyecto robótico y que puedan compartirlo con el resto del mundo.

Una vez presentado podré continuar añadiendo mejoras y muchas cosas que tengo pensadas y que, posiblemente, se realicen para el proyecto del máster de Ingeniería de Sistemas y Computación que me encuentro realizando en la actualidad.

8.3. Mejoras futuras

La aplicación puede mejorarse en diversos aspectos. A continuación, se citan algunas de las mejoras que pueden llevarse a cabo:

- Incorporación de advertencias acústicas tras la detección de una señal de tráfico.
- Mejora del diseño de la interfaz gráfica.
- Permitir la definición de las teclas de control del teclado e incorporación de dispositivos tales como gamepads o joysticks.
- Elaborar un generador de código para la exportación en los dispositivos robóticos, reduciendo por tanto las labores de programación.

Anexos

Anexos con las instrucciones para la instalación de todos los componentes software empleados en el desarrollo del proyecto.

.1. Instalación de Node.js

Instalación de los prerequisites:

```
1 sudo apt-get install python-software-properties python g++ make
```

Si está utilizando Ubuntu 12.10, necesitará hacer lo siguiente:

```
1 sudo apt-get install software-properties-common
```

Añadimos el repositorio:

```
1 sudo add-apt-repository ppa:chris-lea/node.js
```

Actualizamos la lista de paquetes:

```
1 sudo apt-get update
```

Instalación de Node.js:

```
1 sudo apt-get install nodejs
```

.2. Instalación Sails.js

Esta guía proporciona las pautas necesarias para la configuración de un entorno de trabajo para el desarrollo de aplicaciones Sails. Esta guía no cubre la instalación de un entorno de producción.

.2.1. Prerrequisitos

Partiendo de que se encuentra Node correctamente instalado en una máquina con Ubuntu 16.04.2 LTS. Ubuntu es una plataforma muy popular y utilizada en el desarrollo de Sails.js, al igual que otros sistemas operativos basados en Unix, como Mac OS X. La instalación es relativamente fácil y existe multitud de información gracias a su amplia comunidad de desarrolladores.

- Una máquina con Ubuntu 16.04.2 LTS
- Node js instalado.

.2.2. Instalación

A continuación detallaremos los pasos para la instalación de Sails. Lo primero que haremos es instalar Sails haciendo uso de npm, el gestor de paquetes que viene con el propio Node. Para ello, lo que vamos a hacer es ir directamente a la terminal y e introducir lo siguiente:

```
1 sudo npm install sails -g
```

La opción -g, que significa global, lo que hace es instalar Sails a nivel global, la cual nos permitirá acceder a las funcionalidades de Sails que emplearemos para la creación de nuestros proyectos.

Es posible que necesite permisos de administrador para instalar Sails a nivel global.

Para comprobar que la instalación se realizó correctamente, crearemos un proyecto inicial y levantaremos el servidor. Para ello introducimos en la terminal:

```
1 sails new my_first_app
```

Cambie de directorio (cd) al nuevo directorio creado, en el cual nos aseguraremos de que Sails está correctamente instalado. Para ello arrancaremos el servidor y comprobaremos en nuestro navegador en localhost 1337 (localhost: 1337) que Sails está funcionando correctamente.

```
1 cd my_fisrt_app
2 sails lift
```

Tras introducir en nuestro navegador *localhost: 1337* debemos obtener el siguiente resultado:

A terminal window with a dark background and light green text. The prompt is 'manuel@manuel-ThinkPad-E560:~/Escritorio/my_first_app:ruby-2.3.1\$'. The command 'sails lift' has been executed. The output shows 'Info: Starting app...' followed by 'Info: Sails v0.12.11' and a decorative ASCII art logo. Below the logo, it says 'Info: Server lifted in "/>

Figura 1: Iniciando Sails ¹.

Y en nuestro navegador:

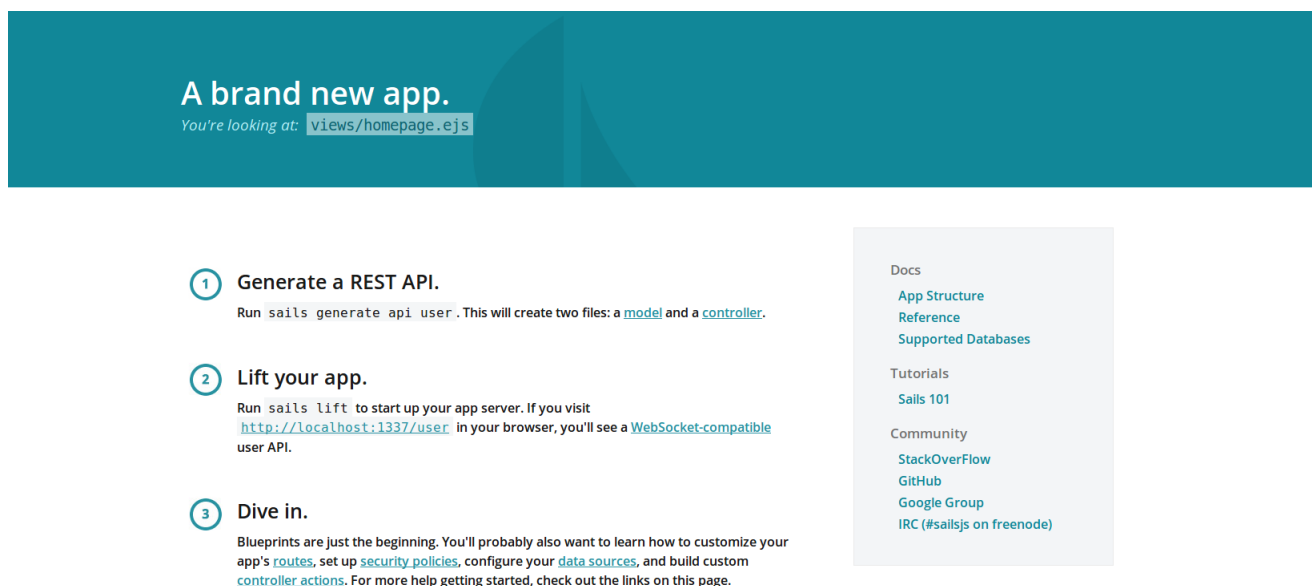


Figura 2: Sails en funcionamiento ².

.3. Instalación de MongoDB

MongoDB es una base de datos libre y de código abierto NoSQL utilizada comúnmente en aplicaciones web modernas. Esta guía le ayudará a configurar MongoDB en su máquina para un entorno de aplicación de producción.

.3.1. Prerrequisitos

Para seguir esta guía es necesario:

- Una máquina con Ubuntu 14.04.
- Un usuario con permisos de administrador (no root).

.3.2. Instalación

MongoDB ya está incluido en los repositorios de paquetes de Ubuntu, pero el repositorio oficial MongoDB proporciona la versión más actualizada y es la forma recomendada de instalar el software. Ubuntu garantiza la autenticidad de los paquetes de software al verificar que están firmados con las claves GPG, por lo que primero tenemos que importar la clave para el repositorio oficial de MongoDB.

Para ello, ejecute:

```
1 sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
  7FOCEB10
```

Después de importar con éxito la clave, verá lo siguiente:

```
1 Gpg: N mero total procesado: 1
2 Gpg: importado: 1 (RSA: 1)
```

A continuación, tenemos que actualizar los detalles del repositorio de MongoDB para que APT sepa de dónde descargar los paquetes.

Introduzca el siguiente comando para crear un archivo de lista para MongoDB.

```
1 echo "deb http://repo.mongodb.org/apt/ubuntu \"$(lsb_release
  -sc)"/mongodb-org/3.0 multiverse" | sudo tee
  /etc/apt/sources.list.d/mongodb-org-3.0.list
```

Después de agregar los detalles del repositorio, necesitamos actualizar la lista de paquetes.

```
1 sudo apt-get update
```

Ahora podemos instalar el propio paquete MongoDB.

```
1 sudo apt-get install -y mongodb-org
```

Este comando instalará varios paquetes que contengan la última versión estable de MongoDB junto con útiles herramientas de administración para el servidor MongoDB.

Después de la instalación del paquete, MongoDB se iniciará automáticamente. Puede comprobarlo ejecutando el siguiente comando.

```
1 service mongodb status
```

Si MongoDB se está ejecutando, verá una salida como la siguiente (con un ID de proceso diferente).

```
1 mongodb.service - An object/document-oriented database
2   Loaded: loaded (/lib/systemd/system/mongodb.service; enabled;
        vendor preset: enabled)
3   Active: active (running) since s b 2017-02-04 13:07:01 CET; 11h ago
4     Docs: man:mongod(1)
5  Main PID: 807 (mongod)
6    Tasks: 10
7   Memory: 84.8M
8      CPU: 4min 7.494s
9   CGroup: /system.slice/mongodb.service
10          - 807 /usr/bin/mongod --config /etc/mongodb.conf
```

También puede detener, iniciar y reiniciar MongoDB utilizando los siguientes comandos

```
1 service mongodb stop
2 service mongodb start
```

.4. Instalación de WebStorm

.5. Robomongo

.6. Git

.6.1. Prerrequisitos

Para seguir esta guía es necesario:

- Una máquina con Ubuntu 14.04.

.6.2. Instalación

La forma más sencilla de disponer Git instalado y listo para su utilización es mediante los repositorios predeterminados de Ubuntu. Este es el método más rápido, pero la versión puede ser más antigua que la versión más reciente. Si necesita la última versión, deberá seguir los pasos para compilar git desde el origen.

Puede utilizar las herramientas de administración de paquetes de apt para actualizar su índice de paquetes local. Después, puede descargar e instalar. Para ello introduce los siguientes comandos:

```
1 sudo apt-get update
2 sudo apt-get install git
```

Esto descargará e instalará git en su sistema. A continuación tendrá que completar los pasos de configuración.

.6.3. Configuración

Ahora que tiene git instalado, necesita hacer algunas cosas para que los mensajes de confirmación que se generarán para usted contendrán su información correcta.

La forma más sencilla de hacerlo es a través del comando git config. Específicamente, necesitamos proporcionar nuestro nombre y dirección de correo electrónico porque git incorpora esta información en cada commit que hacemos. Podemos seguir adelante y agregar esta información escribiendo:

```
1 git config --global user.name "Su nombre"
2 git config --global user.email "tuemail@dominio.com"
```

Podemos ver todos los elementos de configuración que se han establecido escribiendo:

```
1 git config --list
```

Configuración git

User.name = Su nombre User.email=youremail@domain.com

Hay muchas otras opciones que puede establecer, pero estos son los dos esenciales necesarios.

.7. Despliegue

Primaremanet, debemos definir el entorno de producción para nuestra aplicación Sails. Para ello editamos el fichero '/config/env/production.js' de la siguiente manera:

```

1  /**
2  * Production environment settings
3  *
4  * This file can include shared settings for a production environment,
5  * such as API keys or remote database passwords. If you're using
6  * a version control solution for your Sails app, this file will
7  * be committed to your repository unless you add it to your .gitignore
8  * file. If your repository will be publicly viewable, don't add
9  * any private information to this file!
10 *
11 */
12
13 module.exports = {
14
15   /*****
16    * Set the default database connection for models in the production *
17    * environment (see config/connections.js and config/models.js )     *
18    *****/
19
20   models: {
21     connection: 'MongoDBServer',
22   },
23
24   /*****
25    * Set the port in the production environment to 80                  *
26    *****/
27
28   port: 1337,
29
30   /*****
31    * Set the log level in production environment to "silent"          *
32    *****/
33
34   log: {
35     level: "silent"
36   },
37
38   proxyHost: '46.101.102.33',
39   proxyPort: 1337
40
41 };

```

En el servidor de producción se ha utilizado Nginx³ con la siguiente configuración:

```

1
2 server {

```

³Nginx es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3)


```
3  listen      80;
4  server_name 46.101.102.33;
5
6
7  listen 443 ssl;
8
9  ssl_certificate /etc/nginx/ssl/nginx.crt;
10 ssl_certificate_key /etc/nginx/ssl/nginx.key;
11
12 location / {
13     proxy_pass      http://46.101.102.33:1337/;
14
15     proxy_http_version 1.1;
16     proxy_set_header Upgrade $http_upgrade;
17     proxy_set_header Connection "upgrade";
18     proxy_set_header Host $host;
19
20 }
21
22 }
```

Para arrancar la aplicación, se ha hecho uso del gestor de procesos Pm2, se introduce la siguiente orden:

```
1  sudo pm2 start app.js -x -- --prod
```

Para monitorizar la aplicación, Pm2 incorpora una herramienta de gestión de procesos. Para visualizarlo introducimos en una terminal:

```
1  sudo pm2 monit
```

Obteniendo un resultado similar al siguiente:



Figura 3: Utilización del gestor de procesos Pm2 en el entorno de producción.

Bibliografía

- [1] Douglas Crockford. *JavaScript: The Good Parts*. O'Reilly, 2008.
- [2] Digital Ocean. Digital Ocean. <https://www.digitalocean.com/>.
- [3] Git Hub. Git Hub. <https://github.com>.
- [4] Marc Harter T.J. Holowaychuk Nathan Rajlich Mike Cantelon, Alex R. Young. *Node.js in Action*. Manning Publications, 2017.
- [5] Irl Nathan Mike McNeil. *Sails.js in Action*. Manning Publications, 2017.
- [6] Métrica v3. Métrica v3. <http://www.csi.map.es/csi/metrica3>.
- [7] Irl Nathan. Activityoverlord, an application to learn sails.js. <https://github.com/irlnathan/activityoverlord>.
- [8] JJ Merelo Pablo Hinojoda. *Aprende Git: ... y, de camino, GitHub*. 2015.
- [9] Rohit Rai. *Socket.IO Real-Time Web Application Development*. Packt, 2013.
- [10] Sails.js. Sails.js | Realtime MVC Framework for Node.js. <http://sailsjs.com/>.
- [11] Wikibooks. The Book of LaTeX. <http://en.wikibooks.org/wiki/LaTeX>.

GNU Documentation Free License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

[<http://fsf.org/>](http://fsf.org/)

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “**Preserve**

the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the

“History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects. You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and

disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for

anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with ... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.