



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería Informática

Sistema de Navegación Semiautónomo en Interiores



Presentado por Mario Bartolomé Manovel
en Universidad de Burgos — 3 de febrero de 2018

Tutores: Dr. Alejandro Merino Gómez
Dr. César Ignacio García Osorio
Dr. José Francisco Díez Pastor



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Dr. Alejandro Merino Gómez, profesor del departamento de Ingeniería Electromecánica, área de Ingeniería de Sistemas y Automática. D. Dr. César Ignacio García Osorio y D. Dr. José Francisco Díez Pastor, profesores del departamento de Ingeniería Civil, área de Lenguajes y Sistemas Informáticos.

Exponen:

Que el alumno D. Mario Bartolomé Manovel, con DNI 71298657Z, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado «Sistema de Navegación Semiautónomo en Interiores».

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 3 de febrero de 2018

Vº. Bº. del Tutor:

Vº. Bº. del Tutor:

Vº. Bº. del Tutor:

D. Dr. Alejandro
Merino Gómez

D. Dr. César Ignacio
García Osorio

D. Dr. José Francisco
Díez Pastor

Resumen

El objetivo del proyecto es diseñar un sistema de navegación semi-autónomo en espacios cerrados, destinado a la asistencia en vigilancia de seguridad mediante drones. Dada una estancia, el drone deberá ser capaz de realizar un recorrido por el interior, grabando vídeo que será emitido a un servidor, y transmitiendo su posición dentro del mapa a un responsable de seguridad.

Descriptores

drone, UAV, semi-autónomo, semi-automático, vigilancia, navegación, interior, vídeo, filtro de partículas, campos potenciales, búsqueda de ruta, MSP, raspberry pi

Abstract

The point of this project is to design a semi-autonomous navigation system in enclosed spaces, destined to aid security vigilance using drones. Given an enclosed space, the drone should be able to make its path through it, recording video which will be streamed to a server, and updating its position inside of it to the security guard in charge.

Keywords

drone, UAV, semi-autonomous, semi-automatic, vigilance, navigation, indoor, video, particle filter, potential fields, path searching, MSP, raspberry pi

Índice general

Índice general	III
Índice de figuras	IV
Índice de tablas	V
Introducción	1
Objetivos del proyecto	3
Conceptos teóricos	5
3.1. Algoritmia	5
3.2. Dispositivos Físicos	6
3.3. Protocolos	11
3.4. Referencias	13
3.5. Imágenes	13
3.6. Listas de items	14
3.7. Tablas	15
Técnicas y herramientas	17
Aspectos relevantes del desarrollo del proyecto	19
Trabajos relacionados	21
Conclusiones y Líneas de trabajo futuras	23

Índice de figuras

3.1. Raspberry Pi 3.	7
3.2. Benchmark de lector microSD OC.	8
3.3. Controladora de Vuelo Flip32.	9
3.4. Modulación en Ancho de Pulso. Arriba 100 % del ciclo usado. Centro 25 % del ciclo usado. Abajo 50 % del ciclo usado	11
3.5. Composición de un mensaje MSP	12
3.6. Autómata para una expresión vacía	14

Índice de tablas

3.1. Componentes de una Raspberry Pi 3 Model B	6
3.2. Herramientas y tecnologías utilizadas en cada parte del proyecto . .	15

Introducción

En este documento se encuentra toda la información relacionada con el Trabajo de Fin de Grado titulado *Sistema de Navegación Semiautónomo en Interiores*.

En él se puede encontrar la siguiente información:

- **Conceptos teóricos:** Ofrecen una base teórica de la que partir, para llevar a cabo el desarrollo completo del proyecto.
- **Técnicas y herramientas:** Se trata de las implementaciones de, ó uso dado a, los distintos conceptos teóricos anteriormente descritos.
- **Aspectos relevantes del desarrollo del proyecto:** Proporciona información detallada que se ha tenido en cuenta durante las diferentes fases de desarrollo del proyecto.
- **Trabajos relacionados:** Se trata de una lista, junto con una breve descripción, de los diferentes proyectos, papers o trabajos relacionados con el proyecto llevado a cabo.
- **Conclusiones y líneas de trabajo futuras:** Detalla una serie de posibles mejoras, modificaciones e incluso derivaciones, que pueden surgir del proyecto realizado.

Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.

Conceptos teóricos

Esta sección auna los diferentes conocimientos teóricos necesarios para la realización del proyecto. A continuación, y en este orden, se explicarán los algoritmos utilizados, protocolos de comunicación y sistemas físicos empleados.

3.1. Algoritmia

Filtros de Partículas

Los Filtros de Partículas son modelos utilizados para tratar de estimar el estado de un sistema que cambia con el tiempo.

Fue definido como *bootstrap filter* en 1993 por N. Gordon, D. Salmond y A. Smith. Se trata de un método que pretende implementar filtros bayesianos recursivos, haciendo uso del método de Montecarlo, es decir, realiza repetidas medidas del estado para estimar la posición del sistema.

Subsubsecciones

Y subsecciones.

Campos Potenciales

A la hora de lograr una navegación segura en un entorno determinado, es necesario implementar un sistema de evasión de obstáculos.

El algoritmo de Campos Potenciales aporta una manera de evitar colisionar con los diferentes obstáculos existentes.

Componente	Raspberry Pi 3B
CPU	BCM2837
Núcleos	4
Velocidad	1.2GHz
RAM	1GB
Coms	Ethernet, WiFi, Bluetooth
USB	4 (2.0)
GPIO	40
Consumo máximo	6.7W

Tabla 3.1: Componentes de una Raspberry Pi 3 Model B

3.2. Dispositivos Físicos

Raspberry Pi

Una Raspberry Pi es un pequeño ordenador desarrollado en UK por la Raspberry Pi Foundation, con la intención de promover el aprendizaje de informática básica en colegios y países en desarrollo. El modelo base se compone de una única placa de medidas 85mm x 56mm (LxA), y unos 42g de peso.

Concretamente el modelo empleado es una Raspberry Pi 3B, y consta de los componentes detallados en la tabla 3.1

Su reducido tamaño y bajo consumo lo hacen ideal para este tipo de proyecto. En este caso se utiliza bajo una distribución GNU/Linux llamada Raspbian¹, basada en Debian. Para tratar de mejorar su rendimiento y reducir al mínimo el consumo, se ha escogido la versión Lite del sistema, es decir, un sistema mínimo sin entorno de escritorio y con la mayor parte de servicios desactivados por defecto.

Una vez descargado el sistema, este ha sido instalado en una micro SD mediante el siguiente procedimiento por consola²:

- `diskutil list` Permite localizar el dispositivo en el que se encuentra la tarjeta. En nuestro caso `/dev/disk4`
- `diskutil umountDisk /dev/disk4` Permite desmontar el volumen.
- `sudo dd if=raspbian-stretch.img of=/dev/rdisk4 bs=1m` El comando `dd` copia la entrada estándar a la salida estándar. Mediante `if/of` se establece el fichero de entrada/salida. Mediante `bs` se establece el

¹Descargable desde: <https://www.raspberrypi.org/downloads/raspbian>

²Realizado en OSX, aunque en Linux es muy similar

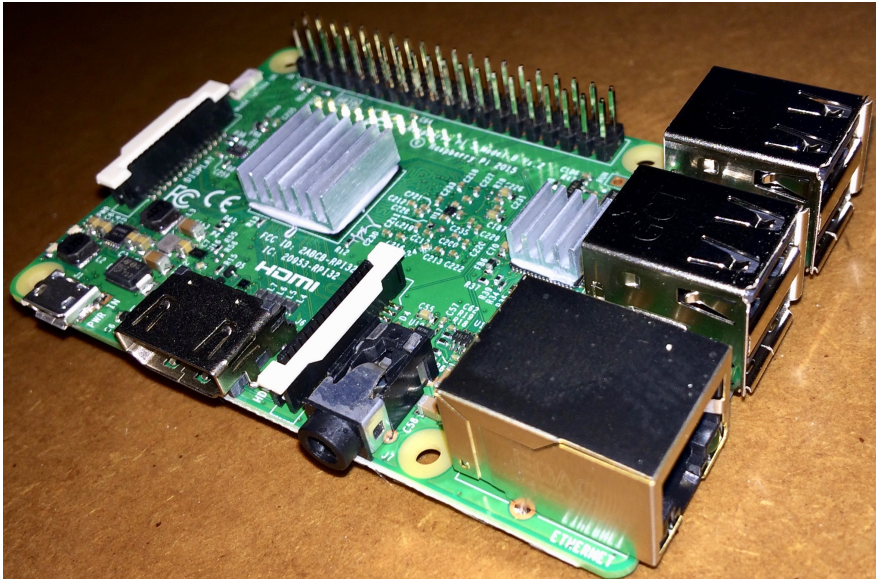


Figura 3.1: Raspberry Pi 3.

tamaño de bloque a copiar. Se está utilizando `/dev/rdisk4` en lugar de `/dev/disk4` debido a la capacidad de OSX de trabajar con dispositivos en bruto, *raw*, de forma que es posible acceder al dispositivo de forma directa³, sin almacenar en un buffer la lectura del archivo, proporcionando velocidades de escritura/lectura hasta 20 veces más rápidas.

Una vez realizados estos pasos, se puede insertar la microSD en la Raspberry Pi. Para encenderla basta con utilizar el puerto micro-usb de que dispone. La Raspberry Pi 3 requiere de una fuente de alimentación capaz de proporcionar 2,5A⁴ para funcionar al máximo nivel de estrés para el procesador y alimentar dispositivos USB.

Sin embargo, este no es estrictamente nuestro caso, véase el apartado **Modificaciones**. Se requiere un dispositivo cuyo consumo sea lo más reducido posible, pero que sea rápido en la ejecución, y que muestre poca latencia en operaciones de `IO`, que es donde se encuentra el cuello de botella.

Una vez encendida, se accede a ella con el usuario por defecto `pi` y la contraseña por defecto `raspberrypi`. Obviamente ambas **han sido cambiadas** por motivos de seguridad.

³Véase `man hdiutil`, sección *DEVICE SPECIAL FILES*

⁴Véase <https://www.raspberrypi.org/help/faqs/#power>

```

CONFIG:
CLOCK : 100.000 MHz
CORE  : 400 MHz, turbo=0
DATA  : 512 MB, /root/test.dat

HDPARM:
=====
Timing 0_DIRECT disk reads: 108 MB in 3.02 seconds = 35.78 MB/sec
Timing 0_DIRECT disk reads: 108 MB in 3.02 seconds = 35.72 MB/sec
Timing 0_DIRECT disk reads: 108 MB in 3.02 seconds = 35.75 MB/sec

WRITE:
=====
536870912 bytes (537 MB, 512 MiB) copied, 21.7232 s, 24.7 MB/s
536870912 bytes (537 MB, 512 MiB) copied, 20.7831 s, 25.8 MB/s
536870912 bytes (537 MB, 512 MiB) copied, 21.6338 s, 24.8 MB/s

READ:
=====
536870912 bytes (537 MB, 512 MiB) copied, 14.3944 s, 37.3 MB/s
536870912 bytes (537 MB, 512 MiB) copied, 14.3785 s, 37.3 MB/s
536870912 bytes (537 MB, 512 MiB) copied, 14.3567 s, 37.4 MB/s

```

Figura 3.2: Benchmark de lector microSD OC.

Modificaciones

- Se ha desactivado el puerto HDMI para reducir el consumo en ~30mA:
Para ello se ha incluido en `/etc/rc.local` la línea `/usr/bin/tvservice -o`.
- Se ha overclockeado el lector de microSD a 100MHz, en lugar de los 50MHz por defecto:
Para ello se ha incluido en `/boot/config.txt` la línea `dtoverlay=sd_overclock=100`.
Y que arroja los resultados mostrados en la imagen 3.2
- Se han incluido una serie de disipadores para evitar sobrecalentamiento de la placa. Así como un pequeño ventilador de bajo consumo.

Controladora de Vuelo

Una controladora de vuelo (*FC* de aquí en adelante) es un pequeño circuito integrado, que contiene un procesador, una serie de sensores, y una serie de entradas y salidas. La FC se encarga de mantener el sistema de estabilización del dron, tomando medidas de los sensores de que dispone, tales como un acelerómetro, giroscopio, magnetómetro, barómetro... etc.

En el caso de la FC usada para este proyecto, llamada Flip32 y mostrada en la imagen 3.3, se dispone de un IC MPU-6050, en rojo, con acelerómetro y giroscopio, así como de un barómetro M55611, en verde, y un magnetómetro HMC5883L, en amarillo. El núcleo de esta pequeña placa es un procesador STM32F103, en cyan, a 72MHz y basado en arquitectura ARM el cual dispone

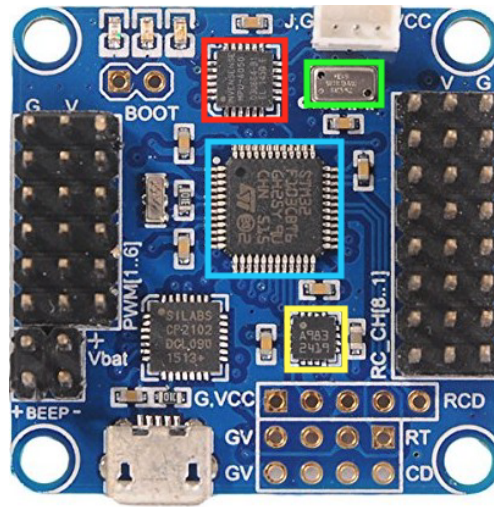


Figura 3.3: Controladora de Vuelo Flip32.

de dos puertos serie, que permiten establecer comunicación entre la FC y otros dispositivos, como emisoras u otros sensores.

El puerto micro-USB de que dispone es utilizado para establecer comunicación entre el configurador de opciones del drone, o en el caso de nuestro proyecto, para hacer uso del [MultiWii Serial Protocol](#).

Entradas

En el lado derecho de la imagen 3.3 pueden verse una serie de pines que actúan como 8 canales de entrada desde el receptor de radio. Dichos canales de entrada, por defecto, reciben una señal modulada en ancho de pulso o **PWM**

Salidas

En el lado izquierdo de la imagen 3.3, pueden verse otros pines que actúan como salida de señal hacia los controladores de velocidad de los motores del drone (ESC o *Electronic Speed Controller*). Estos pines de salida, emiten una señal que será interpretada por los ESC del drone, para determinar la frecuencia dada al voltaje que alimenta los motores. En el caso de nuestro proyecto, los ESC disponibles reciben una señal PWM, al igual que la FC desde un receptor de radio.

Sensores

La controladora de vuelo Flip32 en su versión más completa, dispone de los siguientes sensores:

- IMU⁵ MPU-6050: Se trata de un circuito integrado compuesto de un acelerómetro de tres (3) ejes y un giroscopio de tres (3) ejes. El acelerómetro mide las fuerzas en los tres diferentes ejes (en g), el giroscopio se encarga de medir la velocidad angular en cada uno de los tres ejes (en $\text{deg}^\circ/\text{s}$)
- Magnetómetro HMC5883L: Se trata de un pequeño magnetómetro digital capaz de medir el campo magnético terrestre (en Gauss). Se debe tener en cuenta que según la posición en el planeta, el campo magnético varía entre $G0,25$ - $G0,65$, así como la declinación magnética de la zona en la que se realiza la medición.⁶
- Barómetro M55611: Se trata de un altímetro de alta precisión, con resoluciones de hasta 10cm, que funciona midiendo la presión atmosférica (en milibar). Hay que tener en cuenta que los cambios de presión, como los generados por las hélices del drone, hace variar la medida del sensor. Por ello, en el caso de usarlo, se cubre con un pequeño filtro de un material absorbente.

⁵Unidad de Medición Inercial.

⁶Disponible en: <http://magnetic-declination.com/>

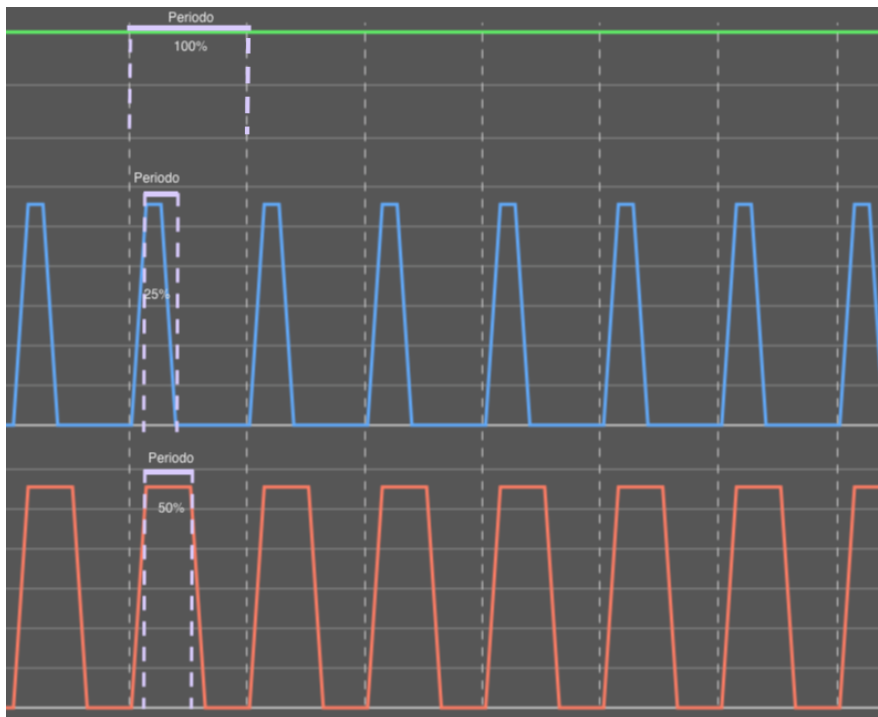


Figura 3.4: Modulación en Ancho de Pulso. Arriba 100 % del ciclo usado. Centro 25 % del ciclo usado. Abajo 50 % del ciclo usado

3.3. Protocolos

Pulse Width Modulation

La modulación por ancho de pulso, se basa en medir el transcurso de tiempo entre el flanco de subida de una señal, y el flanco de bajada, tal y como puede verse en la imagen 3.4. En este contexto, un receptor de radio recibe una señal de la emisora, y genera una señal PWM acorde que será transmitida a la FC. Estas son capaces de entender señales de entre 1000 y 2000 μ s. Cualquier valor por debajo, o por encima, haría entrar la controladora en FailSafe⁷ En el caso de este proyecto, se ha determinado que no se hará uso de este protocolo para la comunicación entre la Raspberry Pi y la controladora de vuelo, véase ?? y 3.3, y por lo tanto el uso de estos pines queda descartado.

Sin embargo, la comunicación entre la FC y los ESC se realiza mediante este tipo de señal, por ello se ha considerado relevante explicar, brevemente, su funcionamiento.

⁷Al recibir una señal inválida por parte del receptor, la controladora de vuelo puede ser configurada para desactivar el drone, mantener la última medida buena conocida, intentar aterrizar... etc. Este modo es conocido como FailSafe

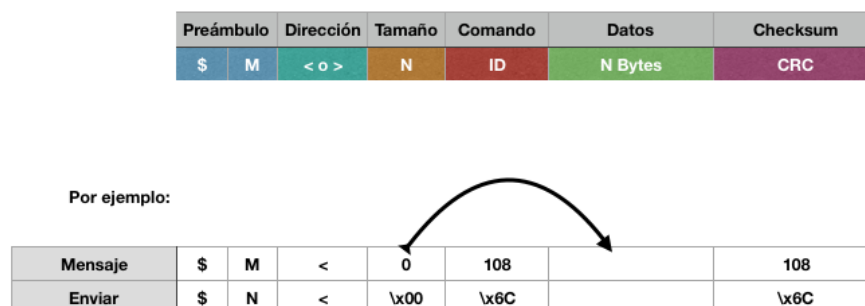


Figura 3.5: Composición de un mensaje MSP

MultiWii Serial Protocol

MultiWii es un software de control de multirrotores y está basado en componentes de la Nintendo Wii. Los mandos de control de la consola de Nintendo poseen tres acelerómetros para determinar la posición angular y medir aceleraciones laterales. El problema es que los acelerómetros no son precisos para variaciones pequeñas, de forma que Nintendo creó un complemento que se podía conectar al propio mando, el Wii Motion Plus, que dispone de tres giroscopios, de forma que unidos a los tres acelerómetros iniciales, proporcionan una medición mucho más precisa de la posición del mando.

Para crear un entorno amigable para el usuario común, se desarrolló una aplicación de escritorio capaz de configurar los parámetros de esta controladora de vuelo poco convencional. De alguna forma debía lograrse una comunicación entre la FC y la aplicación, y así se implementó MultiWii Serial Protocol. Conocido como *MSP*, se trata de un protocolo que se ha mantenido en diferentes implementaciones de sistemas de control de vuelo. No solo aquellos basados en sensores de la Nintendo Wii controlados por Arduino, sino en otros más modernos y potentes que han ido surgiendo los últimos años. Como el utilizado en nuestro proyecto.

Se trata de un protocolo eficiente y sencillo, que permite una comunicación rápida y completa. Su composición puede verse en la imagen [3.5](#)

- **Preámbulo:** Se trata de dos caracteres ASCII que determinan el comienzo de un nuevo mensaje. Se compone de un símbolo '\$' y una letra 'M'.
- **Dirección:** Se trata de un carácter ASCII que determina la dirección del mensaje, **hacia** la controladora '<' o **desde** la controladora '>'.
- **Tamaño:** Se trata de un byte que determina el tamaño, en bytes, de los datos enviados o recibidos.

- Comando: Se trata de un byte que determina el comando a ejecutar. Véase ?? para una relación de los comandos implementados.
- Datos: Se trata de una serie de bytes, de longitud definida por el byte $\langle tamaño \rangle$, que establecen el resto de parámetros a pasar con la función definida por el byte $\langle comando \rangle$.
- Checksum: Se trata de un byte de control. Se define su valor mediante la función XOR entre $\langle tamaño \rangle$, $\langle comando \rangle$ y cada byte contenido en $\langle datos \rangle$

En el caso de nuestro proyecto, se hará uso de este protocolo dado que existe la posibilidad de establecer la recepción de los canales de radio a través de un puerto serie. Es decir, en lugar de utilizar un receptor de radio, se utilizará un puerto serie para obtener la telemetría⁸, y establecer las entradas de los canales de radio.

3.4. Referencias

Las referencias se incluyen en el texto usando cite [?]. Para citar webs, artículos o libros [?].

3.5. Imágenes

Se pueden incluir imágenes con los comandos standard de L^AT_EX, pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:

⁸Se define telemetría como un sistema de medición de magnitudes a distancia; en este contexto el término se ha desvirtuado y se entiende como la información que es transmitida de vuelta a la emisora de vuelo, siendo esta generalmente información de los sensores de la FC, GPS o voltaje disponible en la batería.



Figura 3.6: Autómata para una expresión vacía

3.6. Listas de items

Existen tres posibilidades:

- primer item.
 - segundo item.
1. primer item.
 2. segundo item.

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

▪

3.7. Tablas

Igualmente se pueden usar los comandos específicos de L^AT_EX o bien usar alguno de los comandos de la plantilla.

Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket		X	X	X	X
MikT _E X					X
T _E XMaker					X
Astah					X
Balsamiq Mockups		X			
VersionOne		X	X	X	X

Tabla 3.2: Herramientas y tecnologías utilizadas en cada parte del proyecto

Técnicas y herramientas

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros³, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.