**UNIVERSIDAD DE MÁLAGA**

**ESCUELA TÉCNICA SUPERIOR DE**

**INGENIERÍA DE TELECOMUNICACIÓN**





**PROYECTO FIN DE CARRERA**

*TÍTULO DEL PFC*

**INGENIERÍA DE TELECOMUNICACIÓN**

MÁLAGA, 2005 MANUEL JESÚS ROMERO PERALES

**ESCUELA TÉCNICA SUPERIOR DE**

**INGENIERÍA DE TELECOMUNICACIÓN**

**UNIVERSIDAD DE MÁLAGA**

**Titulación: Ingeniería de Telecomunicación**

Reunido el tribunal examinador en el día de la fecha, constituido por:

D./Dª.

D./Dª.

D./Dª.

para juzgar el Proyecto Fin de Carrera titulado:

**TÍTULO DEL PFC**

del alumno D./Dª.

dirigido por D./Dª.

ACORDÓ POR \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ OTORGAR LA CALIFICACIÓN DE \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Y, para que conste, se extiende firmada por los componentes del tribunal, la presente diligencia

Málaga, a \_\_\_\_\_\_ de \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ de \_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| El Presidente | El Vocal | El Secretario |
|  |  |  |
| Fdo.: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Fdo.: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Fdo.: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**ESCUELA TÉCNICA SUPERIOR DE**

**INGENIERÍA DE TELECOMUNICACIÓN**

**UNIVERSIDAD DE MÁLAGA**

**TÍTULO DEL PFC**

**REALIZADO POR:**

*Manuel Jesús Romero Perales*

**DIRIGIDO POR:**

*Ricardo Ron Ángevin*

**DEPARTAMENTO DE: Tecnología Electrónica**

**TITULACIÓN: Ingeniería de Telecomunicación**

**PALABRAS CLAVE:**

**RESUMEN:**

Málaga, mes y año de presentación

Agradecimientos

Esta sección no es en absoluto obligatoria, pero es el lugar correcto para dedicar el proyecto a las personas/instituciones/empresas/… que se desee.

Resumen

Un texto breve (una cara aproximadamente) que describa qué se ha hecho en el proyecto, sus principales objetivos, la utilidad que se le quiere dar, si está destinado a algún cliente real, aspectos sobre la tecnología usada y cosas similares que permitan hacerse una idea rápida del trabajo realizado.

Se trata de describir brevemente todos los aspectos más importantes del proyecto destacando en lo posible sus puntos fuertes para permitir comprenderlo fácilmente en una lectura rápida sin tener más referencias del mismo. Por tanto, no debe ser un texto demasiado largo ni complejo.

Palabras Clave

Palabra1, Palabra2, Palabra3, …

De 5 a 7 palabras[[1]](#footnote-2) clave que mencionen conceptos de capital importancia en el proyecto: Cosas que el proyecto manipula (Ej.: “Máquinas Expendedoras”, “Automóviles”), tecnologías usadas (Ej.: J2EE, RMI), utilidad del proyecto (Ej.: Gestión de Existencias), temática (Ej.: Historia Medieval, Aeronáutica) y cosas similares.

Si finalmente saliesen demasiados términos, conviene hacer una selección de los más relevantes para quedarse con el número indicado.

*Abstract*

Traducción al inglés del resumen anterior. Conviene hacerlo una vez se tenga la versión definitiva de dicho resumen. Se recomienda consultar al director del proyecto acerca de si considera adecuado que aparezca esta sección.

*Keywords*

Ídem a la sección anterior. Se recomienda consultar al director del proyecto acerca de si considera adecuado que aparezca esta sección.

Índice General

[1 Introducción 19](#_Toc228290788)

[1.1 Motivaciones del proyecto 19](#_Toc228290789)

[1.2 Objetivos 19](#_Toc228290790)

[1.3 Alcance del proyecto 21](#_Toc228290791)

[1.4 Estudio de la Situación Actual 21](#_Toc228290792)

[1.4.1 Evaluación de Alternativas 22](#_Toc228290793)

[2 Aspectos teóricos relevantes 23](#_Toc228290794)

[2.1 Sistema BCI (Interfaces Cerebro-Computadora) 23](#_Toc228290795)

[2.2 Realidad Virtual 23](#_Toc228290796)

[2.3 VRML (Virtual Reality Modelling Language) 23](#_Toc228290797)

[3 Herramientas software utilizadas 24](#_Toc228290798)

[3.1 3D Studio Max 24](#_Toc228290799)

[3.2 Cortona 3D Viewer 24](#_Toc228290800)

[3.3 Matlab 24](#_Toc228290801)

[4 Descripción general de los mundos 25](#_Toc228290802)

[5 Modelado de Vivienda Virtual 26](#_Toc228290803)

[5.1 Descripción general 26](#_Toc228290804)

[5.2 Implementación 26](#_Toc228290805)

[5.2.1 Plano de planta 26](#_Toc228290806)

[5.2.2 Paredes y contorno a partir del plano 26](#_Toc228290807)

[5.2.3 Planos de suelo 26](#_Toc228290808)

[5.2.4 Huecos para puertas y ventanas 27](#_Toc228290809)

[5.2.5 Puertas 27](#_Toc228290810)

[5.2.6 Ventanas 27](#_Toc228290811)

[5.2.7 Exteriores 27](#_Toc228290812)

[5.2.8 Iluminación de la escena 27](#_Toc228290813)

[5.2.9 Texturización 28](#_Toc228290814)

[5.2.10 Decoración interior 29](#_Toc228290815)

[5.3 Integración con sistema BCI existente 29](#_Toc228290816)

[5.3.1 Sensores de proximidad 29](#_Toc228290817)

[5.3.2 Escalado de dimensiones 29](#_Toc228290818)

[5.3.3 Exportación a VRML 29](#_Toc228290819)

[6 Modelado de ETSIT 31](#_Toc228290820)

[6.1 Descripción general 31](#_Toc228290821)

[6.2 Implementación 31](#_Toc228290822)

[7 Modelado del simulador de Vuelo 32](#_Toc228290823)

[7.1 Descripción general 32](#_Toc228290824)

[7.2 Implementación 32](#_Toc228290825)

[8 Conclusiones y Ampliaciones 33](#_Toc228290826)

[8.1 Conclusiones 33](#_Toc228290827)

[8.2 Ampliaciones 33](#_Toc228290828)

[9 Referencias Bibliográficas 35](#_Toc228290829)

[9.1 Libros y Artículos 35](#_Toc228290830)

[9.2 Referencias en Internet 36](#_Toc228290831)

[10 Apéndices 37](#_Toc228290832)

[10.1 Glosario y Diccionario de Datos 37](#_Toc228290833)

[10.2 Contenido Entregado en el CD-ROM 38](#_Toc228290834)

[10.2.1 Contenidos 38](#_Toc228290835)

[10.2.2 Código Ejecutable e Instalación 40](#_Toc228290836)

[10.2.3 Ficheros de Configuración 40](#_Toc228290837)

[10.3 Índice Alfabético 41](#_Toc228290838)

[10.4 Código Fuente 42](#_Toc228290839)

[10.4.1 Paquete Ejemplo 1: 42](#_Toc228290840)

Índice de Figuras

[Figura 1‑1: Interior de vivienda virtual 20](#_Toc228290785)

[Figura 1‑2: ETSIT Telecomunicación virtual 20](#_Toc228290786)

[Figura 1‑3: Simulador virtual de vuelo 21](#_Toc228290787)

# Introducción

En los siguientes capítulos de esta memoria se va a describir el diseño e implementación de diferentes entornos virtuales desarrollados utilizando técnicas de realidad virtual

El concepto de memoria de un proyecto es, en esencia, un resumen del proyecto para personas que desconozcan o no posean conocimientos avanzados de la naturaleza del proyecto y/o sus tecnologías, o incluso no posean conocimientos específicos de informática. Por tanto, debemos orientarla de manera que cualquier persona pueda entender que se ha hecho durante todo el proyecto.

Los puntos obligatorios de la memoria varían mucho de unos proyectos a otros, pero en este documento se proponen unos mínimos. En muchos casos, la memoria tiene un apartado por cada parte importante del proyecto, por ejemplo (Introducción, Requerimientos, Análisis y Diseño, Presupuesto, etc.) y en cada apartado se resume (para el perfil de lector mencionado anteriormente) el contenido del apartado técnico correspondiente. En cualquier caso, podemos orientar la memoria de la siguiente forma:

## Motivaciones del proyecto

La principal motivación del desarrollo de este proyecto es la de proporcionar a los sistemas BCI de entornos virtuales que se asemejen, con el mayor grado posible a la realidad, con la finalidad de realizar de una manera mucho más entretenida y familiar el uso de estos sistemas, y que el paso a su implementación y explotación en entornos reales sea menos costosa.

## Objetivos

El objetivo de este proyecto es el diseño e implementación de mundos tridimensionales virtuales, utilizando técnicas de realidad virtual, logrando el mayor grado de inmersión posible, proporcionando la sensación visual de encontrarse en los tres entornos siguientes:

1. El interior de una vivienda.
2. La planta baja de la Escuela Técnica Superior de Telecomunicaciones de Málaga.
3. Un simulador de vuelo, recreando el vuelo sobre la ciudad de Málaga y alrededores.

Un último objetivo es el de adecuar el desarrollo e implementación de los mundos virtuales de modo que la integración con el sistema BCI existente y el interfaz de navegación, elaborado en el Departamento de Tecnología Electrónica, se produzca de forma rápida y sencilla .



Figura ‑: Interior de vivienda virtual



Figura ‑: ETSIT Telecomunicación virtual

En el caso del simulador de vuelo, mediante el interfaz de navegación gobernaremos un avión que sobrevuela la ciudad de Málaga.

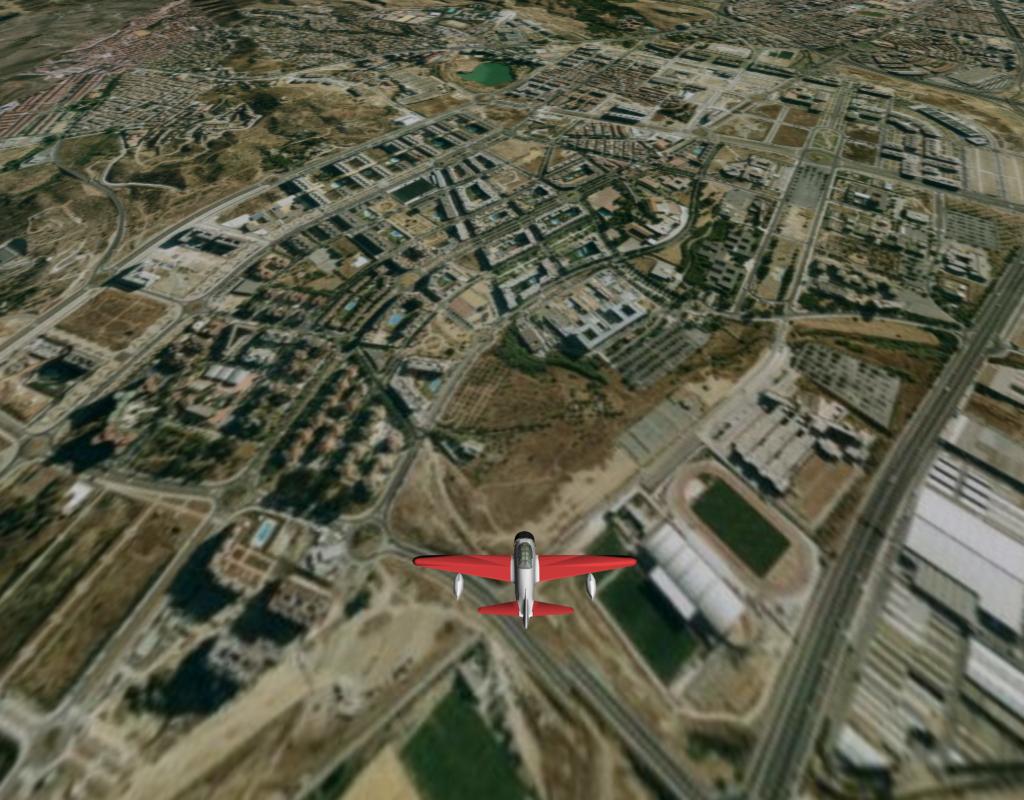


Figura ‑: Simulador virtual de vuelo

El primer mundo virtual nos sumerge en un entorno muy usual para cualquier individuo como es el interior de una vivienda estándar, en el que podemos encontrar los recintos más habituales, una entrada, un salón, un dormitorio, baño y terraza. Se trata de una sola planta dado que el interfaz de navegación es una silla de ruedas cuya restricción principal es la de no subir escaleras, por tanto podría asemejarse a la vivienda de un individuo con la imposibilidad de mover sus extremidades, de ahí que las puertas tengan suficiente tamaño y los espacios abiertos sean lo suficientemente amplios.

El segundo mundo virtual desarrollado nos hace vivir la experiencia de un autentico “paseo virtual” por la Escuela de Técnica Superior de Telecomunicaciones de Málaga. Este entorno es altamente conocido por todo aquel que haya cursado alguna de las carreras que en este centro se imparten. Del mismo modo, se desarrolla exclusivamente la planta baja por el mismo motivo que en el mundo anterior, dado que igualmente se integra con el sistema BCI existente que introduce la silla de ruedas junto con el interfaz de navegación.

## Alcance del proyecto

## Estudio de la Situación Actual

En esta sección deben identificarse y describirse sistemas similares al que se va a desarrollar, estableciendo una comparación entre lo que ofrecen estos sistemas y lo que pretendemos lograr con el proyecto, para de esta forma diferenciar nuestro desarrollo de lo ya existente.

No tienen porque ser sistemas que hagan lo mismo que el nuestro, sino que pueden ser sistemas que contengan funcionalidad en común con una parte significativa o bien que estén orientados a un conjunto de potenciales usuarios similar.

En esta sección también es adecuado evaluar las posibles herramientas o lenguajes de programación utilizables para el proyecto y determinar cual (o cuales) se adaptan mejor a nuestras necesidades concretas (de forma justificada).

Conviene en general destacar los puntos en común y las principales discrepancias entre estos sistemas y el nuestro, con la idea de ver en qué sentido nuestro desarrollo supone una ganancia o mejora sobre ellos (también puede orientarse a resolver ciertos defectos de los mismos, mejorar algunas funciones para hacerla más completa, rápida o fácil de usar, etc.). Si los sistemas carecen de alguna funcionalidad que el nuestro va a incorporar, conviene también destacarlo (precisamente esto puede ser una de las principales aportaciones del mismo).

En general, conviene usar esta sección como un primer paso para “promocionar” las “bondades” de nuestro proyecto.

### Evaluación de Alternativas

En esta sección se describirán, una por una, todas las alternativas estudiadas. Conviene estudiar 3 o 4 alternativas importantes, salvo que por algún motivo justificado se deba incluir un número menor o mayor de las mismas. En todo caso, siempre es conveniente cuidar de que en esta sección haya un conjunto de sistemas significativo. En función de lo dicho anteriormente, cada sistema podrá dividirse en tres secciones: “Descripción”, “Ventajas” e “Inconvenientes”, aunque es posible cualquier otra división que contenga los aspectos descritos, dependiendo de qué tipo de sistemas se estudien.

#### Sistema 1

#### Sistema 2

# Aspectos teóricos relevantes

## Sistema BCI (Interfaces Cerebro-Computadora)

Definición

## Realidad Virtual

Existe un gran número de conceptos asociados con las palabras *realidad virtual.* Pero el que más se aproxima a la intención que persigue este proyecto es la siguiente:

*"Realidad virtual: un sistema de computación usado para crear un mundo artificial en el que el usuario tiene la impresión de estar en ese mundo y la habilidad de navegar y manipular objetos en él". Manetta C. y R. Blade (1995)*

Por lo tanto, la realidad virtual es la representación de una serie elementos a través de software informático y equipamiento electrónico, con el objetivo de crear entornos sintéticos con el mayor realismo posible en los que sumergir al individuo, proporcionándole la sensación de encontrarse realmente en el mundo y con la capacidad de navegar e interactuar con los elementos que se encuentran dentro de él.

DESCRIBIR LOS TIPOS DE REALIDAD VIRTUAL y SOBRETODO ESPECIFICAR MAS CONCIENZUDAMENTE LA REALIDAD VIRTUAL INMERSIVA

## VRML (Virtual Reality Modelling Language)

VRML stands for Virtual Reality Modelling Language. VRML is an ASCII-based open, non-proprietary language. This means that it can be used by anyone without licensing. It has been officially adopted by International Standardization Organization (ISO). The current version of this language supports animation, spatial sound, collision detection and scripting. Virtual reality is defined by a VRML file which consists of a multi-tree of nodes. Each node is of a predefined type but note that new types of nodes can be created.

The Virtual Reality aspect of VRML is centred on the metaphor which it pursues: human space. That space is 3-dimensional and defines the ways we move in it, perceive it and interact with it. VRML accordingly includes many of the things that are required in making the virtual world: a way of describing the geometry which creates the objects and spaces we move around in - light, texture and sound. We can approach and view theobjects from different angles. We can hear the sound from different positions.

The second part of VRML, Modelling Language, describes the process of making a virtual world. While much of VRML is composed and optimised in text, VRML authors typically use other modelling programs for designing and creating 3D geometry in a graphical format. Later, these models, which will constitute a Virtual World, are translated to VRML and viewed using a VRML browser. The browser renders the VRML in real time and turns the code into perceivable space with which we can interact.

**3ds max** supports the modelling process in two ways:

1. with a modelling environment capable of producing high quality 3D models

2. with VRML helper objects

Helper objects are very important in VRML worlds. Without them, the user is still able to move around the virtual world but interaction is limited to looking at objects from different angles. Integration of sound, touch and other spatial perceptions can only be accomplished by using helper objects - which significantly extend interactivity.

# Herramientas software utilizadas

## 3D Studio Max

## Cortona 3D Viewer

## Matlab

# Descripción general de los mundos

# Modelado de Vivienda Virtual

## Descripción general

## Implementación

### Plano de planta

Se introduce un plano de planta con las dimensiones adecuadas al mundo que se quiere implementar. Este plano se texturiza con la imagen de la planta del edificio, una casa en este caso.

### Paredes y contorno a partir del plano

Utilizando el plano de planta como plantilla para el inicio del desarrollo y extrusión de la casa en 3 dimensiones, se van creando las paredes de la casa.

Para ello se utiliza la herramienta WALL del menú XXXXXXX. Con esta herramienta se dibujan las paredes, con formando las habitaciones y departamentos de los que consta la casa.

Con esta herramienta se van creando las paredes, que al fin y al cabo con cajas o boxes de una determinada altura, anchura y largura.

Una caja en 3D Studio se un conjunto de planos que conforman la forma de la caja. Es útil saber que los planos vienen determinados por su posición y un conjunto de normales que definen hacia donde el plano es visible. De esta manera un caja será un conjunto de 4 planos cuyas normales apuntan hacia el exterior de la caja.

El interior de las cajas (paredes en este caso) es hueco, no existiendo elementos macizos en 3d Studio.

### Planos de suelo

A cada habitáculo creado se le incorpora un plano de suelo individual. De esta manera obtenemos las habitaciones de de la casa virtual. Situamos un plano por cada habitáculo para tener independizados los suelos de cada habitación de manera que la posterior texturización de los suelos la podamos hacer de manera individualizada e independizada, pudiendo utilizar texturas diferentes para cada suelo de cada habitación.

### Huecos para puertas y ventanas

Para realizar los huecos donde más tarde sitúan puertas y ventanas se han utilizado los objetos booleanos.

Estos objetos tienen la capacidad de hacer operaciones booleanas. Es decir podemos realizar las operaciones de substracción, unión e intersección con elementos definidos con el modificador BOOLEAN.

De esta manera se realizan los huecos de las paredes y ventanas.

A las paredes anteriormente implementadas se le intersecan cajas (elementos geométricos tipo box) en las localizaciones donde se quieren realizar los huecos y con las dimensiones diseñadas para cada ventana o puerta.

Ya solo nos queda realizar la substracciones pertinentes, es decir substraer a las paredes las cajas intersecadas, conformando los huecos.

### Puertas

Se utilizan objetos de 3D Studio que nos crean puertas tipo que más tarde se puden modelar y caracterizar a gusto del diseñador.

*Como es un objeto PIVOT DOOR y como se coloca, dibuja. Como se abre la puerta. Etc.*

### Ventanas

Ídem para las ventanas

### Exteriores

*Describir como es el exterior de la casa virtual (terraza) y que se ve desde ahí fuera. Como se ha realizado el cielo. Objetos árboles. Línea horizonte de árboles. ETC.*

### Iluminación de la escena

Número y disposición de los puntos de luz. Por que se ha realizado de la manera escogida?

Efectos de sombra. Explicar como pasa la luz a través delos objetos en 3dStudio (Buscar en internet).

Describir los tipos de luces de 3dStudio válidos para la exportación a WRML.

Tipo de luz escogida en tal caso.

**Plano de techo** de manera que dejar pasar la luz de los puntos de luz hacia el interior y que desde dentro se siga viendo el techo y no el cielo!!!!!.

### Texturización

Texturización de paredes y suelos

Texturización de puertas

Texturización de las ventanas

### Decoración interior

Descripción de los elementos de decoración para cada habitación.

Con el objetivo de realizar la escena lo más real posible.

Descarga desde sites gratuitos y licencia.

Explicar por qué no se ha decidido a implementar desde cero los objetos de decoración y sin embargo se han descargado desde Internet.

Al ser objetos descargados tienen un número de vértices y polígonos muy elevado. Explicar como se han optimizado los modelos para reducir el número de vértices y no sobrecargar la escena.

Re texturización de algunos elementos de decoración.

Objetos de decoración nuevos: cortinas, objetos con demasiados vértices que han sido necesarios re implementarlos con muchos menos vértices.

Objetos que se han utilizado:

Sanitarios de los baños: lavabo, bidet, bañera, váter, espejo.

Salón: Sofá, sillones, mesa y 4 sillas, televisión, marcos para cuadros, mesita, cortinas, mueble de la televisión con repisas.

Dormitorio: cama, cómoda.

Terraza: baranda, columpio mecedora.

## Integración con sistema BCI existente

### Sensores de proximidad

### Escalado de dimensiones

### Exportación a VRML

# Modelado de ETSIT

## Descripción general

## Implementación

# Modelado del simulador de Vuelo

## Descripción general

## Implementación

# Conclusiones y Ampliaciones

## Conclusiones

Conclusiones del sistema: Qué hemos elaborado, si los resultados están dentro de lo esperado, si hemos cumplido las expectativas, justificación de haber escogido las mejores opciones para cada uno de los aspectos del sistema, etc.

## Ampliaciones

Cualquier labor de ampliación que tengamos contemplada en el sistema debe ser descrita aquí, mencionando en qué consiste, cómo ampliará el sistema, qué ventajas nos aporta y porqué no se ha incluido en el sistema diseñado, entre otros aspectos.

# Referencias Bibliográficas

## Libros y Artículos

Libros y artículos usados de alguna forma durante el desarrollo del proyecto o su documentación.

**Formato sugerido:**

**[<PrimerApellidoAutor><DosUltimosDigitosDelAño>]** <Apellidos1, Nombre1; Apellidos2, Nombre2;…>. ”<Título del libro o Articulo>”. <Editorial o lugar de publicación>. <Año (4 cifras)>.

**Ejemplo:**

**[Redondo07]** Redondo L., J. Manuel; De Tal y Cual, Menganito. ”Ejemplo para la plantilla de PFC”. Universidad de Oviedo. 2007.

Si tenemos el ISBN, debemos también ponerlo al final.

## Referencias en Internet

Páginas Web consultadas para cualquier aspecto relacionado con el desarrollo del sistema o su documentación.

**Formato sugerido:**

**[<PrimerApellidoAutor><DosUltimosDigitosDelAño>]** <Apellidos1, Nombre1; Apellidos2, Nombre2;…>. “<Título de la página Web>”. <URL>. <Año en el que se consultó (4 cifras)>.

**Ejemplo:**

**[Redondo07]** Redondo L., J. Manuel; De Tal y Cual, Menganito. “Título de la página Web de ejemplo”. www.unaurlcualquiera.com. 2007.

Si tenemos más datos que permitan localizar la información dentro de la página, podemos ponerla donde consideremos oportuno.

Esta referencia es real (se usa dentro del documento) y debe dejarse aquí siempre que usemos el cuestionario que la menciona en la sección de usabilidad.

**[Hassan08]** Hassan Montero, Y. “Guía de Evaluación Heurística de Sitios Web”. <http://www.nosolousabilidad.com/articulos/heuristica.htm>

# Apéndices

## Glosario y Diccionario de Datos

Por orden alfabético, todos los términos que se consideren importantes en la aplicación con una descripción breve de su significado dentro de la aplicación.

* **Término1**: Descripción del significado.
* **Término2**: Descripción del significado.

## Contenido Entregado en el CD-ROM

### Contenidos

Descripción del contenido de los diskettes o *CDs* (directorios y para qué sirve cada cosa), descripción de esta documentación y de cualquier material que adicionalmente se entregue en la presentación. En esta sección se presenta una estructura de directorios de ejemplo para el CD que se puede seguir para distribuir todos sus contenidos en el mismo. Conviene por tanto tenerla en cuenta desde el principio de la implementación.

##### Introducción

La estructura de directorios del proyecto debe poder recoger todos los ficheros relacionados con el proyecto, clasificándolos por su propósito dentro del mismo. Los tipos más frecuentes son: ficheros fuente, ficheros de configuración, ficheros de documentación…

Se deben crear directorios para contener cada uno de los tipos de ficheros. Tener una estructura estandarizada de los directorios del proyecto es importante por varias razones:

* Ayuda a localizar la información del proyecto. Por ejemplo, los ficheros fuente siempre deben estar en la carpeta *src*.
* Ayuda a los desarrolladores a determinar donde debe ir cada fichero.
* Permite crear scripts de construcción estandarizados.

##### Recomendación estructura general directorios del CD

**NOTA:** El nombre del CD debe corresponder con el del proyecto.

|  |  |
| --- | --- |
| **Directorio** | **Contenido** |
| *./ Directorio raíz del CD* | Contiene un fichero leeme.txt explicando toda esta estructura. Se puede incluir *autorun* e icono del proyecto si existe. |
| *./<nombre\_proyecto>* | Contiene toda la estructura de directorios del proyecto para desarrollo. **Ver la tabla Recomendación de estructura de directorios de desarrollo**. <nombre\_proyecto> debe sustituirse por el nombre corto del proyecto. |
| *./instalacion* | Ficheros utilizados para la instalación del proyecto. |
| *./documentacion* | Contiene toda la documentación asociada al proyecto. Es necesario incluir un fichero con el documento final del proyecto (en formato .*doc* o .*docx* de *Word* o bien .*sxw* de *Open Office*, por ejemplo) además de un fichero .*PDF* |
| *./documentacion/img* | Directorio que contiene las imágenes utilizadas en la documentación. Estas imágenes tendrán formato .*png* si son capturas de pantalla, .*wmf* si son diagramas o esquemas y .*jpg* sólo si son fotografías. |
| *./documentacion/uml* | Ficheros que genera la herramienta (Rose, *ArgoUML*, etc.) con la que se han generado los diagramas UML y de entidad relación. |
| *./presentacion* | Directorio que contiene la presentación en *Powerpoint* o equivalente utilizada el día de la defensa del proyecto, si está disponible en el momento de realizar el CD. |
| *./herram* | Contiene los ficheros de instalación de las herramientas utilizadas para el desarrollo o puesta en marcha del proyecto (lógicamente sólo las que sean distribuibles). |
| *./herram/desarrollo* | Ficheros de instalación de las herramientas utilizadas en el desarrollo |
| *./herram/explotacion* | BD, servidor Web y herramientas en general. |

##### Recomendación de la Estructura de Directorios de “desarrollo”

Se muestra aquí el contenido del directorio de desarrollo de la tabla anterior, incluyendo todos los directorios que deben depender del mismo. Algunos de los elementos sea han incluido suponiendo que se están usando ciertas tecnologías Java. En caso de no usarlas, buscaremos un equivalente existente (si lo hay) en la que estemos usando nosotros.

|  |  |
| --- | --- |
| **Directorio** | **Contenido** |
| *./ Directorio raíz de “desarrollo”* | Contiene los ficheros de proyecto del IDE utilizado. |
| *./build* | Contiene el build.xml de *ant* (si lo usamos). Debemos situarnos dentro para poder invocarlo. |
| *./conf* | Contiene los diferentes ficheros de configuración del proyecto. Podría contener distintos subdirectorios, en función de la tecnología usada. En este ejemplo se muestra un ejemplo de un proyecto *Web* hecho con tecnologías *Java*:   * **web**: contiene los ficheros de configuración de la aplicación Web (por ejemplo: web.xml). * **ear**: contiene los ficheros de configuración de una aplicación empresarial (por ejemplo: application.xml). * **ejb**: contiene los descriptores de despliegue de los EJB. |
| *./dist* | Directorio donde se sitúan los ficheros para la distribución del proyecto. Por ejemplo: los ficheros *.war* o *.ear.* |
| *./doc* | Contiene toda la documentación relativa al proyecto, incluyendo los ficheros generados por herramientas de generación de documentación automática como *Javadoc* o similar. |
| *./lib* | Bibliotecas externas (.*jar*, .*dll*, …) necesarias para compilar y distribuir, de las que depende este proyecto. |
| *./compile-lib* | Bibliotecas externas (.*jar*, .*dll*, …) necesarias para compilar pero que no deseamos distribuir. |
| *./src* | Ficheros fuente |
| *./src/java* | Todos los ficheros *Java*, lógicamente agrupados en los paquetes correspondientes. |
| *./src/sql* | Este directorio contiene los scripts de *SQL* que permiten construir y meter los datos iniciales en la base de datos del proyecto (si existe). |
| *./web* | Este directorio contiene los ficheros (.JSP, .ASPX, .HTML, …) de la Web (si el proyecto incluyese una). |
| ./web/images | Contiene las imágenes utilizadas por los ficheros de la web del proyecto. |
| ./classes | Directorio donde se guardan los ficheros compilados (como por ejemplo los .*class*) |
| ./test | Directorio base para todos los ficheros utilizados en la automatización del proceso de prueba. |
| ./test/java | Contiene todas las pruebas unitarias utilizadas en el proceso de prueba automatizado. |
| ./test/sql | Scripts *SQL* utilizados en la carga de datos de prueba |
| ./bak | Directorio donde se pueden guardar versiones antiguas de los ficheros fuente del proyecto. |

### Código Ejecutable e Instalación

Descripción de los contenidos del código ejecutable y de la instalación de la aplicación en un ordenador. Breve manual de instalación y puesta en marcha de la aplicación (solamente unos pasos sencillos que faciliten este proceso, sin explicar nada (para eso está el manual de instalación propiamente dicho).

### Ficheros de Configuración

Descripción de todos los ficheros necesarios para poder hacer funcionar la aplicación (ficheros de configuración, ficheros de datos, etc.).

## Índice Alfabético

Este índice alfabético esta generado automáticamente por Word e incluirá todos los términos que nosotros marquemos adecuadamente con la herramienta que Word posee para ello. No debemos pues incluir palabras “a mano” en él. Este índice tiene una serie de ejemplos para ilustrar como funciona. No debemos olvidarnos de usar la opción “Actualizar campos” al finalizar de la documentación.

**NOTA: Quitar esta explicación en la documentación final.**

I

índice alfabético, 103

P

Palabra1, 7

problemas encontrados, 64

R

Redondo L., J. Manuel, 98

## Código Fuente

El código fuente tiene que ir dividido por paquetes y por archivos, con un formato que haga que resulte legible, tal y como se muestra a continuación en este ejemplo. En la mayoría de las ocasiones, es posible que no sea conveniente colocar la totalidad del código fuente en esta sección, sino solo una parte, que contenga aquellas clases más significativas e importantes. En cualquier caso, conviene consultar al director del proyecto este aspecto. Para dar un formato al código apropiado, podemos probar a copiarlo directamente de nuestro editor y ver si el texto acarrea sus propiedades de estilo (no funcionará en todos los editores), o bien usar programas como *Scintilla* (*Windows*) (<http://www.scintilla.org/>) o el editor *Kate* (para *KDE*, *Linux*) (<http://kate-editor.org/>)

### Paquete Ejemplo 1:

#### Fichero “A.cs”:

using System;

using System.Collections.Generic;

using System.Text;

namespace ControlesAvisos

{

public class Usuario: ICloneable

{

public Usuario()

{

this.nombre = "";

this.clave = "";

}

public Usuario(string nombre, string clave)

{

this.nombre = nombre;

this.clave = clave;

}

private string nombre;

public string Nombre

{

get

{

return nombre;

}

set

{

nombre = value;

}

}

private string clave;

public string Clave

{

get

{

return clave;

}

set

{

clave = value;

}

}

public Object Clone()

{

Usuario u = new Usuario(this.Nombre, this.Clave);

return u;

}

}

}

1. No necesariamente debe ser una única palabra, pueden ser varias. Por ejemplo, si el proyecto tratase sobre la gestión de calificaciones de Alumnos, una palabra clave válida podría ser “Expediente Académico”. [↑](#footnote-ref-2)