

UNIVERSIDAD DE SANTIAGO DE COMPOSTELA  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA



# Inteligencia Artificial aplicada a Videojuegos Top-Down en tiempo real

TRABAJO DE FIN DE GRADO

*Autor:*

Rubén Osorio López

*Tutores:*

Manuel Mucientes Molina

Pablo Rodríguez Mier





**D. Manuel Mucientes Molina**, Profesor del Departamento de Electrónica y Computación de la Universidad de Santiago de Compostela, y **D. Pablo Rodríguez Mier**, Profesor del Departamento de Electrónica y Computación de la Universidad de Santiago de Compostela,

INFORMAN:

Que la presente memoria, titulada *Inteligencia Artificial aplicada a Videojuegos Top-Down en tiempo real*, presentada por **D. Rubén Osorio López** para superar los créditos correspondientes al Trabajo de Fin de Grado de la titulación de Grado en Ingeniería Informática, se ha realizado bajo nuestra tutoría en el Departamento de Electrónica y Computación de la Universidad de Santiago de Compostela.

Y para que así conste a los efectos oportunos, expiden el presente informe en Santiago de Compostela a (Data):

El tutor,

El cotutor,

El alumno,

Manuel Mucientes Molina   Pablo Rodríguez Mier   Rubén Osorio López



# Agradecimientos

*A mi tutor Manuel y a mi cotutor Pablo por su inestimable e imprescindible ayuda durante el periodo de desarrollo de este trabajo.*

*A mis compañeros de clase que me han sabido animar cuando más lo necesitaba.*

*A mi familia por apoyarme durante todo este tiempo.*



# Índice general

<b>1. Prefacio</b>	<b>1</b>
1.1. Objetivos . . . . .	2
1.2. Organización del documento . . . . .	3
<b>2. Introducción</b>	<b>5</b>
<b>3. Análisis de requisitos</b>	<b>7</b>
3.1. Identificación de los requisitos del sistema . . . . .	7
3.1.1. Actores . . . . .	8
3.1.2. Casos de uso . . . . .	8
3.1.3. Especificación de requisitos . . . . .	13
<b>4. Gestión del proyecto</b>	<b>35</b>
4.1. Gestión de riesgos . . . . .	35
4.2. Gestión de la configuración . . . . .	35
4.3. Metodología de desarrollo . . . . .	35
4.4. Planificación temporal . . . . .	36
4.5. Análisis de costes . . . . .	36
4.6. Plan de Gestión de las Comunicaciones . . . . .	36
<b>5. Arquitectura y herramientas</b>	<b>37</b>
5.1. Arquitectura del sistema . . . . .	37
5.2. Herramientas utilizadas . . . . .	37
<b>6. Diseño e implementación</b>	<b>39</b>
<b>7. Validación y pruebas</b>	<b>41</b>
<b>8. Valoraciones finales</b>	<b>43</b>
<b>9. Ejemplos</b>	<b>47</b>
9.1. Un ejemplo de sección . . . . .	47
9.1.1. Un ejemplo de subsección . . . . .	47
9.1.2. Otro ejemplo de subsección . . . . .	47

9.2. Exemplos de figuras e cadros . . . . .	48
9.3. Exemplos de referencias á bibliografía . . . . .	48
9.4. Exemplos de enumeracións . . . . .	48
<b>10. Conclusiones y posibles ampliaciones</b>	<b>51</b>
<b>A. Manuais técnicos</b>	<b>53</b>
<b>B. Manuais de usuario</b>	<b>55</b>
<b>C. Licencia</b>	<b>57</b>
<b>Bibliografía</b>	<b>59</b>



# Índice de figuras

3.1. Diagrama de casos de uso . . . . .	10
9.1. Esta é a figura de tal e cal. . . . .	48



# Índice de cuadros

3.1. Matriz de trazabilidad de requisitos contra casos de uso . . . . .	34
9.1. Esta é a táboa de tal e cal. . . . .	48



# Capítulo 1

## Prefacio

Los juegos de lógica o estrategia han sido utilizados como *benchmark* en el campo de la inteligencia artificial desde sus inicios. Pese a que principalmente se han utilizado juegos de mesa o de cartas dadas las facilidades a la hora de representar su modelo, con el avance de la tecnología y la llegada de los videojuegos en ordenador se consideró la posibilidad de que los mismos podrían ser una nueva herramienta con la que experimentar y explorar diferentes técnicas e implementaciones de IA. En la actualidad prácticamente todos los videojuegos que incorporan elementos que simulan comportamiento humano lo hacen con lógica determinista y patrones definidos, sin incorporar ningún tipo de simulación de pensamiento complejo. Algunas de las razones causantes de este fenómeno son:

- La diferencia en la dificultad de implementación entre patrones simples y un agente complejo.
- El hecho de que un agente puede llegar a ser demasiado efectivo a la hora de competir contra un humano.
- Limitaciones en la capacidad de computación restante cuando el sistema está ejecutando un videojuego ya que suelen ser aplicaciones significativamente pesadas y complejas.

Sin embargo, estos problemas son solventables y debemos tener en cuenta los beneficios que se pueden obtener al trabajar con videojuegos reales. En la actualidad, somos capaces de simular grandes mundos con millones de entidades y jugadores reales. Estos entornos comúnmente utilizan simulaciones físicas realistas y trabajan con datos numéricos continuos en lugar de acciones discretas, es decir, una entidad que simule estar viva en dicho entorno se podría mover cara una posición concreta y realizar una determinada acción durante un periodo de tiempo sin necesidad de que ninguno de estos conceptos esté atado a una representación discreta. Nos encontramos el caso contrario con los juegos de mesa o

cartas tradicionales ya que en estos solo existen acciones discretas y simbólicas, no se trata con ningún tipo de estado continuo.

Por estas razones se podría considerar que un videojuego está mucho mas cerca de ser capaz de modelar el mundo real que un juego de mesa, lo que lo hace un banco de trabajo idóneo para investigar y experimentar con Inteligencia Artificial. Técnicas como redes neuronales, computación evolutiva o lógica difusa pueden funcionar en situaciones en tiempo real, continuas y complejas que presentan tanto nuestra realidad como los videojuegos. Por esto se pretende investigar y experimentar sobre cuales de estas técnicas pueden ser de utilidad en este tipo de situaciones para lo cual se implementará un agente que compita con un jugador real en un pequeño videojuego en el que ambos se moverán libremente en un espacio de posiciones continuo y realizarán una serie de acciones ofensivas y defensivas con el objetivo de derrotar a su oponente.

Con este fin, se pretende realizar una implementación de un juego que permita al agente tener acceso a los estados en los que se encuentran el y su competidor así como las acciones que puede realizar en un determinado momento. Todo esto con la finalidad de averiguar cuales de las posibles técnicas de IA pueden ser usadas para lograr un competidor no solo apto pero también justo. Además, esto nos permitirá realizar un entrenamiento en el que el propio agente podrá competir con él mismo para aprender los fundamentos del juego además de competir contra jugadores humanos. Esto nos brinda la posibilidad de determinar como reacciona ante estos y si es capaz de adaptarse a los posiblemente distintos estilos de juego contra los que se tendrá que enfrentar.

## 1.1. Objetivos

El objetivo global es implementar un agente que sea capaz de competir contra un jugador simulando un comportamiento humano para lo cual se realizará una pequeña demostración de un videojuego y la implementación del agente con las técnicas que resulten ser más eficaces a la hora de lograr el mencionado objetivo.

Los sub-objetivos que tendremos que abarcar son los siguientes:

1. **Implementar la demostración del videojuego:** Se necesita una plataforma que permita tanto a un jugador humano como al agente interactuar con el entorno del videojuego siguiendo ambos las reglas que este mismo define.
2. **Crear la librería que implemente nuestro agente con las técnicas escogidas:** Se requiere realizar una implementación del agente utilizando las técnicas de inteligencia artificial que hayamos escogido para simular un comportamiento aparentemente humano.

3. **Realizar el entrenamiento del agente:** Una vez realizada la implementación del agente se necesitará ejecutar un proceso de entrenamiento para que este adquiriera la información necesaria para comportarse adecuadamente en el entorno competitivo que el juego presenta.
4. **Obtener datos sobre las capacidades del agente:** Se deberán obtener datos sobre el comportamiento del agente en el entorno del juego al competir con otras posibles implementaciones del mismo que no incluyan el uso de técnicas de inteligencia artificial.
5. **Analizar los resultados obtenidos** Haciendo una recopilación de la información obtenida durante las etapas de diseño, implementación y obtención de datos se realizará un análisis que resuma lo que ha logrado el agente centrándose en aspectos como el desafío que es capaz de presentar, si es capaz de simular a un humano y la posibilidad de usar implementaciones similares en videojuegos en un futuro.

## 1.2. Organización del documento

La finalidad de este documento es presentar como se han resuelto los objetivos definidos para el proyecto. Para ello se explicarán las diferentes partes que forman el producto final así como las tareas que han sido realizadas a lo largo del proyecto y que han dado lugar al mismo tal y como se presenta.

Completar  
la orga-  
nización  
del docu-  
mento





# Capítulo 2

## Introducción



# Capítulo 3

## Análisis de requisitos

Referenciando directamente al PMBOK se define la recopilación de requisitos como:

*“[...]el proceso que consiste en definir y documentar las necesidades de los interesados a fin de cumplir con los objetivos del proyecto. El éxito del proyecto depende directamente del cuidado que se tenga en obtener y gestionar los requisitos del proyecto y del producto”*

Como bien se indica en la cita anterior, el éxito de un proyecto está estrechamente relacionado con la calidad y precisión empleados en el proceso de recogida, análisis y gestión de los requisitos del mismo. Esto hace que imperiosa la necesidad de llevar a cabo los procesos mencionados con extrema minuciosidad y cuidado.

A Continuación se estudiarán las necesidades del proyecto en términos de funcionalidades y objetivos deseables para así obtener una idea sobre el alcance del mismo. Para ello se realizará la identificación y descripción de los múltiples requisitos que se deberán cumplir, separándolos según su índole y relacionándolos con los casos de uso apropiados.

### 3.1. Identificación de los requisitos del sistema

El primer paso a la hora de realizar correctamente el proceso de análisis de requisitos es dedicar el tiempo necesario para comprender cuales son los objetivos y alcance reales del mismo. Para ello se han llevado a cabo varias reuniones con los tutores previas a la reunión del proyecto en las que se ha discutido cual sería el producto final buscado así como los requisitos que dicho producto tendría que cumplir para estar a la altura de dicho ideal. En sucesivas reuniones iterativas con ambos tutores se ha logrado tener una idea común del alcance a cumplir mediante el refinamiento de los requisitos, buscando la mínima ambigüedad posible en su definición.

### 3.1.1. Actores

El primer paso que se debe afrontar dentro del proceso de análisis de requisitos es la identificación de los actores que intervendrán en la ejecución del sistema de una forma u otra. Un actor representa cualquier tipo de rol interpretado por una persona, dispositivo o sistema externo que tendrá capacidad de interactuar con nuestro producto. En este caso particular, la definición de los actores es sencilla y directa: el único rol que se debe definir es el del **usuario que interacciona con el videojuego** descrito en esta memoria. En aras de hacer una especificación formar realizaremos la descripción de los actores en la siguiente tabla:

ID	Actor-01
Nombre	Jugador
Descripción	Este actor representa al usuario que interaccionará con el videojuego, realizando partidas contra otro jugador o el agente implementado así como ejecutando el proceso de simulación de partidas que el agente usará como proceso de aprendizaje.

### 3.1.2. Casos de uso

Una vez especificados los actores que interactúan con el sistema el siguiente paso es identificar y especificar los diferentes casos de uso. Dichos casos de uso nos permitirán centrarnos luego en los requisitos del sistema partiendo de una base sólida.

Se definirá caso de uso como [2] una secuencia de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable de valor para un actor. De esta forma se pueden utilizar los casos de uso para definir el comportamiento deseado durante la etapa de desarrollo realizando una abstracción que permitirá que los miembros involucrados en el proyecto puedan hacer referencia a ellos sin preocuparse de detalles de bajo nivel. De esta forma se facilitarán discusiones sobre el funcionamiento deseado del proyecto, incluido el análisis de requisitos.

De este mismo modo, lo que se hará a continuación es reconocer las diferentes acciones que pueden llevar a cabo los actores, en este caso solamente uno, sobre el sistema. Haciendo una primera aproximación a esta definición de un modo informal podemos considerar las siguientes acciones: Un usuario deberá ser capaz de seleccionar entre las diferentes acciones a realizar dentro del entorno del juego, lo que dará lugar a la ejecución de otros casos de uso diferentes (**seleccionar la opción a ejecutar dentro del videojuego**). Las diferentes opciones darán lugar

a casos de uso diferentes ya que el usuario podrá competir con otro usuario con el mismo rol ( **competición de jugador humano contra jugador humano**) o competir con el agente entrenado ( **competición de jugador humano contra agente**). Finalmente se podrá hacer que el agente compita con otra versión de sí mismo (**competición de agente contra agente**). Además esta ultima acción se puede llevar a cabo sin que el actor visualice el proceso exponiendo solo los resultados con el fin de llevar a cabo el proceso de aprendizaje del agente.

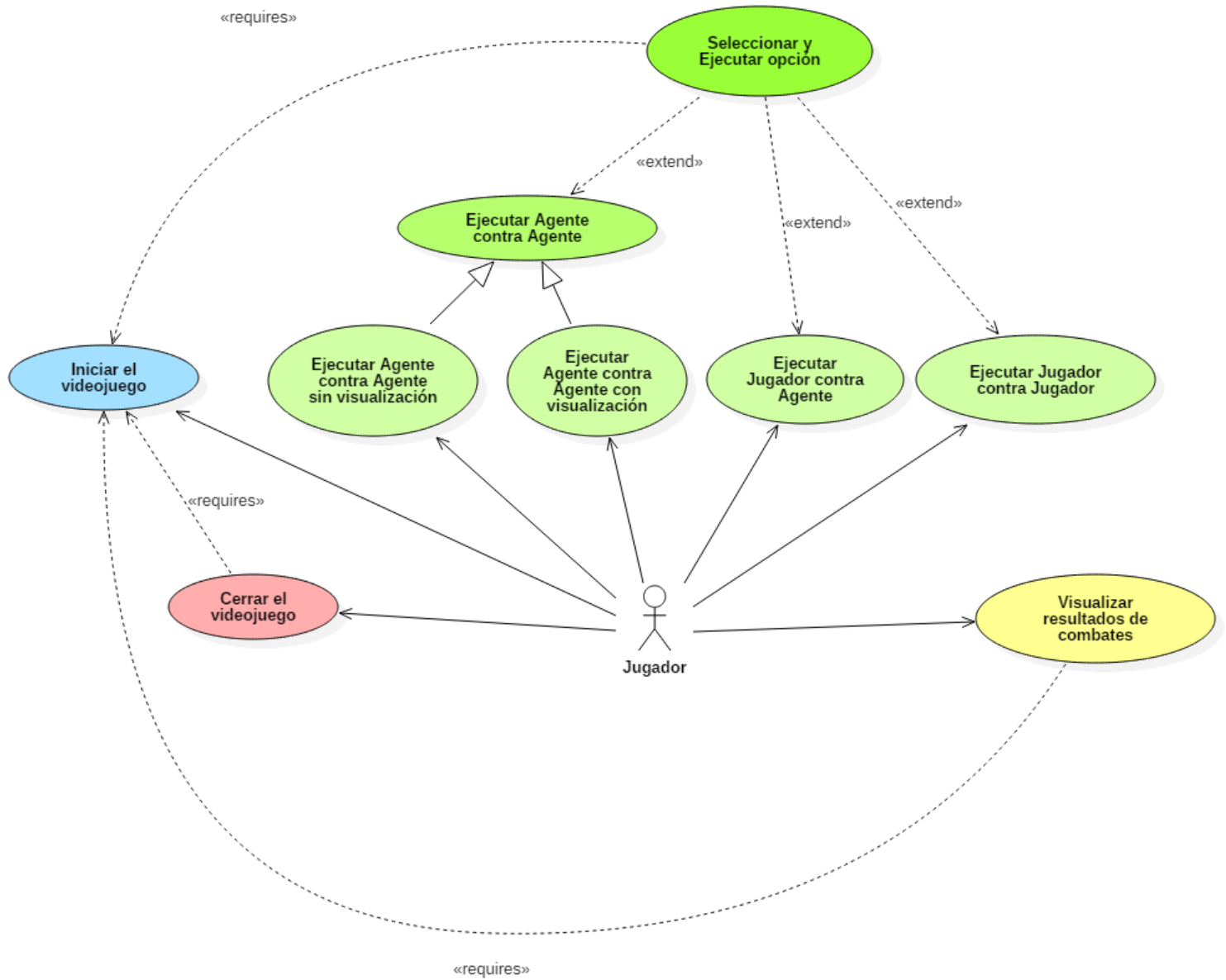


Figura 3.1: Diagrama de casos de uso

En la figura 3.1 se muestra el diagrama de casos de uso a partir del cual podremos realizar una definición formal de los mismos. En dicha figura se han agrupado por colores los casos de uso con características similares para facilitar su comprensión y visualización. En este caso se utilizan el color azul y rojo para abrir y cerrar el programa, colores verdes para los diferentes tipos de ejecuciones posibles del combate y el amarillo para ver los resultados de los combates realizados en una ejecución.

A continuación se muestran los casos de uso especificados de manera formal. La notación utilizada para la identificación de los diferentes casos de uso será  $CU^1-N^2$ . Los campos presentes en cada descripción contarán con un identificador, nombre, descripción y una prioridad. El nivel de prioridad podrá tener uno de los siguientes tres valores con su significado asociado:

- **Vital:** Es fundamental para completar el proyecto.
- **Importante:** Aumenta de forma significativa la calidad y funcionamiento del proyecto pero no tiene la cualidad de indispensable.
- **Deseable:** Extiende el proyecto pero no debería ser una prioridad.

ID	CU-1
Nombre	Iniciar el videojuego
Descripción	El usuario ejecuta el programa.
Prioridad	Vital

ID	CU-2
Nombre	Cerrar el videojuego
Descripción	El usuario cierra el programa.
Prioridad	Vital

---

<sup>1</sup>en referencia a Caso de Uso

<sup>2</sup>en referencia al Número asociado al mismo

<b>ID</b>	CU-3
<b>Nombre</b>	Seleccionar y Ejecutar opción
<b>Descripción</b>	El usuario escoge una de las opciones posibles para ser ejecutada.
<b>Prioridad</b>	Vital

<b>ID</b>	CU-4
<b>Nombre</b>	Ejecutar Agente contra Agente
<b>Descripción</b>	El usuario escoge una de las opciones en las que el agente compite contra otra versión de sí mismo.
<b>Prioridad</b>	Vital

<b>ID</b>	CU-5
<b>Nombre</b>	Ejecutar Agente contra Agente con visualización
<b>Descripción</b>	El usuario escoge la opción de visualizar un combate entre un agente y otra versión de sí mismo.
<b>Prioridad</b>	Vital

<b>ID</b>	CU-6
<b>Nombre</b>	Ejecutar Agente contra Agente sin visualización
<b>Descripción</b>	El usuario escoge la opción de simular un combate entre un agente y otra versión de sí mismo sin realizar una visualización del mismo.
<b>Prioridad</b>	Vital

<b>ID</b>	CU-7
<b>Nombre</b>	Ejecutar Jugador contra Agente
<b>Descripción</b>	El usuario escoge la opción que le permite competir contra el agente.
<b>Prioridad</b>	Vital



<b>ID</b>	CU-8
<b>Nombre</b>	Ejecutar Jugador contra Jugador
<b>Descripción</b>	El usuario escoge la opción que le permite competir contra otro usuario.
<b>Prioridad</b>	Importante

<b>ID</b>	CU-9
<b>Nombre</b>	Visualizar resultados de combates
<b>Descripción</b>	El usuario visualiza los resultados acumulados de los combates que han ocurrido en la ejecución del programa.
<b>Prioridad</b>	Deseable

### 3.1.3. Especificación de requisitos

Una vez definidos los casos de uso se puede proceder a realizar un análisis detenido de cada uno de ellos para así extraer los requisitos que se deben cumplir para hacer posible su ejecución. Esta sección estará dedicada a clasificar y describir con el menor grado de ambigüedad posibles cada uno de los requisitos del proyecto. Para este fin utilizaremos el estándar IEEE830 [3], en el que se define la diferenciación entre tres tipos de requisitos:

- **Requisitos de información:** Identificados con códigos de la forma RI-*N*
- **Requisitos funcionales:** Identificados con códigos de la forma RF-*N*
- **Requisitos no funcionales:** Identificados con códigos de la forma RNF-*N*

Algunos de los campos que se utilizarán para describir los requisitos son comunes, tales como el identificador (ID), nombre, descripción y criticidad. En aras de mantener un buen nivel de consistencia se utilizará una escala similar a la de la importancia de los casos de uso para definir la criticidad, pudiendo tomar la misma los siguientes valores:

- **Vital:** Es fundamental para completar el proyecto.
- **Importante:** Aumenta de forma significativa la calidad y funcionamiento del proyecto pero no tiene la cualidad de indispensable.
- **Deseable:** Extiende el proyecto pero no debería ser una prioridad.

### 3.1.3.1. Requisitos de información

Este subapartado contendrá los requisitos de información del proyecto. Este tipo de requisito especifica los datos que deberán ser almacenados por el sistema para permitir así el cumplimiento de otros requisitos ya sea de forma parcial o total. A continuación se muestran formalmente los requisitos de información del proyecto:

Completar  
requisitos  
asociados  
y sus  
nombres

<b>ID</b>	RI-1
<b>Nombre</b>	Datos aprendidos por el agente
<b>Requisitos asociados</b>	<ul style="list-style-type: none"> <li>▪ RF-5</li> <li>▪ RF-7</li> <li>▪ RF-8</li> <li>▪ RF-9</li> <li>▪ RF-13</li> <li>▪ RF-14</li> </ul>
<b>Descripción</b>	El sistema deberá almacenar los datos que representan el conocimiento aprendido por el agente, sea cual sea su forma.
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>▪ Estados visitados previamente por el agente</li> <li>▪ Acciones tomadas por el agente en un determinado estado</li> <li>▪ Resultado, positivo o negativo, de las acciones tomadas en un estado determinado.</li> </ul>
<b>Criticidad</b>	Vital

<b>ID</b>	RI-2
<b>Nombre</b>	Archivos necesarios para la ejecución del videojuego
<b>Requisitos asociados</b>	Todos los requisitos funcionales están asociados ya que para todas las funcionalidades es necesario tener acceso a estos archivos.
<b>Descripción</b>	El sistema deberá almacenar todos los archivos no relacionados con el código de la aplicación en si misma que se requieren para mostrar los contenidos de la aplicación
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>■ Imágenes que contienen las animaciones de los personajes.</li> <li>■ Archivos de audio que contienen los diferentes sonidos del videojuego.</li> <li>■ Archivos que contienen las diferentes fuentes usadas para mostrar el texto por pantalla.</li> <li>■ Archivos que contienen los "Shaders"<sup>3</sup> utilizados para los diferentes efectos dentro del videojuego.</li> </ul>
<b>Criticidad</b>	Vital

### 3.1.3.2. Requisitos de funcionales

Los requisitos funcionales son utilizados para definir las funciones que un sistema debe ser capaz de realizar, definiendo de esta forma el comportamiento que tendrá el software a nivel interno. La especificación de los mismos muestra como, a partir de su cumplimiento, se llevarán a la práctica los casos de uso definidos para el proyecto.

<b>ID</b>	RF-1
<b>Nombre</b>	Visualizar el menú
<b>Requisitos asociados</b>	<ul style="list-style-type: none"> <li>■ RF-2</li> <li>■ RF-3</li> <li>■ RF-12</li> </ul>
<b>Descripción</b>	El sistema deberá mostrar un menú con las diferentes opciones de combate que es posible ejecutar en la aplicación.
<b>Precondición</b>	Ninguna
<b>Secuencia normal</b>	<ol style="list-style-type: none"> <li>1. El usuario ejecuta la aplicación.</li> <li>2. El usuario visualiza las opciones disponibles distinguiendo la preseleccionada por defecto.</li> </ol>
<b>Secuencia alternativa 1</b>	<ol style="list-style-type: none"> <li>1. El vuelve al menú después de un combate.</li> <li>2. El usuario visualiza las opciones disponibles distinguiendo la preseleccionada por defecto.</li> </ol>
<b>Postcondición</b>	Se pueden ver y distinguir todas las opciones del menú y seleccionar una de ellas.
<b>Criticidad</b>	Vital
<b>Criterio de validación</b>	Se considerará este requisito como aceptado cuando el usuario pueda visualizar correctamente el menú sea cual sea la forma en la que se llegue a él (primera ejecución o después de combate).

<b>ID</b>	RF-2
<b>Nombre</b>	Moverse en el menú
<b>Requisitos asociados</b>	<ul style="list-style-type: none"><li>■ RF-1</li><li>■ RF-3</li><li>■ RF-12</li></ul>
<b>Descripción</b>	El sistema debe permitir viajar entre las diferentes opciones del menú mostrando claramente que opción es la seleccionada actualmente.
<b>Precondición</b>	Se esta visualizando el menú de la aplicación.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario pulsa el botón para moverse en el menú (hacia arriba o abajo).</li></ol>
<b>Postcondición</b>	Se cambia la opción seleccionada a la siguiente o anterior dependiendo del botón pulsado, abajo o arriba respectivamente.
<b>Criticidad</b>	Vital
<b>Criterio de validación</b>	Se considerará este requisito como aceptado cuando el usuario pueda cambiar de opción en el menú utilizando los botones dedicados a ello y se visualice dicho cambio de forma inequívoca.

<b>ID</b>	RF-3
<b>Nombre</b>	Ejecutar una opción del menú
<b>Requisitos asociados</b>	<ul style="list-style-type: none"><li>▪ RF-1</li><li>▪ RF-2</li><li>▪ RF-12</li></ul>
<b>Descripción</b>	El sistema deberá permitir la ejecución de la opción actualmente seleccionada en el menú de la aplicación.
<b>Precondición</b>	Se está visualizando el menú de la aplicación y hay una opción seleccionada.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario pulsa el botón de ejecutar la opción seleccionada del menú.</li></ol>
<b>Postcondición</b>	Se ejecuta la acción seleccionada y se pasa a la escena asociada a dicha acción.
<b>Criticidad</b>	Vital
<b>Criterio de validación</b>	Se considerará este requisito como aceptado cuando el usuario pueda ejecutar todas las opciones del menú y se pase a la escena que se ha asociado con cada una de ellas.

<b>ID</b>	RF-4
<b>Nombre</b>	Alternar apertura de la consola con resultados
<b>Requisitos asociados</b>	<ul style="list-style-type: none"><li>■ RF-4</li></ul>
<b>Descripción</b>	El sistema deberá permitir alternar entre la apertura y cierre de una consola donde se muestren los resultados de los combates previos realizados en esa ejecución del programa.
<b>Precondición</b>	Se han ejecutado combates en la presente ejecución del programa.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario pulsa el botón de abrir/cerrar la consola.</li></ol>
<b>Postcondición</b>	La consola cambia al estado de visualización distinto al que estaba, se cierra si estaba abierta y se abre si estaba cerrada.
<b>Criticidad</b>	Importante
<b>Criterio de validación</b>	Se considerará este requisito como aceptado si la apertura y cierre de la consola de resultados es funcional en cualquier estado posible en el que el usuario pueda visualizar la aplicación.

<b>ID</b>	RF-5
<b>Nombre</b>	Salir de la aplicación
<b>Descripción</b>	El sistema deberá permitir salir de la aplicación sea cual sea el estado en el que se encuentre y sin producir errores ni en la presente ni en futuras ejecuciones.
<b>Precondición</b>	La aplicación está en ejecución en cualquier estado.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario ejecuta la opción del menú dedicada a cerrar la aplicación.</li></ol>
<b>Secuencia alternativa 1</b>	<ol style="list-style-type: none"><li>1. El usuario cierra la ventana de la aplicación haciendo uso de las funcionalidades del sistema de ventanas.</li></ol>
<b>Postcondición</b>	Se ha terminado la ejecución de la aplicación.
<b>Criticidad</b>	Vital
<b>Criterio de validación</b>	Se considerará este requisito como aceptado si el cierre de la aplicación no produce errores sea cual sea el estado actual del programa y usando una de las dos secuencias explicadas.



<b>ID</b>	RF-6
<b>Nombre</b>	Entrar en la escena del combate
<b>Requisitos asociados</b>	<ul style="list-style-type: none"><li>■ RF-3</li></ul>
<b>Descripción</b>	El sistema deberá permitir cambiar a una escena de combate si así lo requiere la ejecución de una de las opciones del menú. Al realizar esta acción se visualizará el combate y se podrá interactuar con el enemigo.
<b>Precondición</b>	Se ha seleccionado una de las opciones del menú que requiere pasar a la escena del combate.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario pulsa el botón de entrar en la escena del combate.</li></ol>
<b>Postcondición</b>	La aplicación se encuentra ahora en una escena de combate.
<b>Criticidad</b>	Vital
<b>Criterio de validación</b>	Se considerará este requisito como aceptado si se entra en una escena de combate al pulsar una opción del menú que lo requiera.

<b>ID</b>	RF-7
<b>Nombre</b>	Moverse en el área de combate.
<b>Requisitos asociados</b>	<ul style="list-style-type: none"><li>▪ RF-8</li><li>▪ RF-9</li></ul>
<b>Descripción</b>	El sistema deberá permitir que el usuario mueva a su personaje en el contexto del combate.
<b>Precondición</b>	La aplicación se encuentra en una escena de combate.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario mueve la palanca o <i>"joystick"</i> en la dirección que quiere mover su personaje.</li></ol>
<b>Postcondición</b>	El personaje se mueve en la dirección indicada por el usuario.
<b>Criticidad</b>	Vital
<b>Criterio de validación</b>	Se considerará este requisito como aceptado si el personaje se mueve hacia la dirección indicada por el usuario.

<b>ID</b>	RF-8
<b>Nombre</b>	Atacar al enemigo
<b>Requisitos asociados</b>	<ul style="list-style-type: none"><li>■ RF-7</li><li>■ RF-9</li></ul>
<b>Descripción</b>	El sistema deberá permitir que un personaje dañe a otro mediante la utilización de un ataque.
<b>Precondición</b>	La aplicación se encuentra en una escena de combate.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario pulsa el botón de atacar.</li></ol>
<b>Secuencia alternativa 1</b>	<ol style="list-style-type: none"><li>1. El agente decide realizar la acción de atacar.</li></ol>
<b>Postcondición</b>	El personaje que debe realizar la acción de atacar comienza la ejecución del ataque.
<b>Criticidad</b>	Vital
<b>Criterio de validación</b>	Se considerará este requisito como aceptado si tanto el usuario como el agente son capaces de atacar cuando deciden realizar esa acción dentro de la escena de combate.

<b>ID</b>	RF-9
<b>Nombre</b>	Defenderse del enemigo
<b>Requisitos asociados</b>	<ul style="list-style-type: none"><li>▪ RF-7</li><li>▪ RF-8</li></ul>
<b>Descripción</b>	El sistema deberá permitir que un usuario realice una acción defensiva contra su contrincante de forma que no pueda recibir daño.
<b>Precondición</b>	La aplicación se encuentra en una escena de combate.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario pulsa el botón de defenderse.</li></ol>
<b>Secuencia alternativa 1</b>	<ol style="list-style-type: none"><li>1. El agente decide realizar la acción de defenderse.</li></ol>
<b>Postcondición</b>	El personaje que debe realizar la acción de defenderse se encuentra en un estado en el que no puede recibir daño.
<b>Criticidad</b>	Vital
<b>Criterio de validación</b>	Se considerará este requisito como aceptado si el personaje se encuentra en un estado de defensa cuando deciden realizar esta acción dentro de la escena de combate.

<b>ID</b>	RF-10
<b>Nombre</b>	Ganar/Perder partida
<b>Requisitos asociados</b>	<ul style="list-style-type: none"><li>■ RF-11</li></ul>
<b>Descripción</b>	El sistema deberá permitir que uno de los personajes gane la partida si la vida de su contrincante llega a cero. Haciendo de forma efectiva que uno de los personajes pierda y otro gane.
<b>Precondición</b>	La aplicación se encuentra una escena de combate.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. Uno de los personajes lleva a cabo una acción que desemboca en que la vida de uno de ellos llegue a cero.</li></ol>
<b>Postcondición</b>	Se muestra por pantalla claramente cual de los personajes ha ganado y se detiene el combate.
<b>Criticidad</b>	Vital
<b>Criterio de validación</b>	Se considerará este requisito como aceptado si siempre que la vida de uno de los personajes llegue a cero se detiene la pelea y se nombre ganador de la misma al personaje superviviente.

<b>ID</b>	RF-11
<b>Nombre</b>	Agotar el tiempo de combate
<b>Requisitos asociados</b>	<ul style="list-style-type: none"><li>■ RF-10</li></ul>
<b>Descripción</b>	La aplicación deberá permitir que un combate termine sin que la vida de uno de los personajes llegue a cero si se termina el tiempo máximo establecido.
<b>Precondición</b>	La aplicación se encuentra en una escena de combate.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario espera a que el tiempo de combate llegue a cero.</li></ol>
<b>Postcondición</b>	Se detiene el combate.
<b>Criticidad</b>	Importante
<b>Criterio de validación</b>	Se considerará este requisito como aceptado si sea cual sea el estado de un combate este se detiene al llegar el tiempo disponible para el mismo a cero.

<b>ID</b>	RF-12
<b>Nombre</b>	Volver al menú
<b>Requisitos asociados</b>	<ul style="list-style-type: none"><li>■ RF-1</li></ul>
<b>Descripción</b>	El sistema deberá permitir volver al menú principal desde la escena de combate.
<b>Precondición</b>	La aplicación se encuentra en una escena de combate.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario pulsa el botón de volver al menú.</li></ol>
<b>Postcondición</b>	Se detiene la pelea y se vuelve al menú de la aplicación.
<b>Criticidad</b>	Importante
<b>Criterio de validación</b>	Se considerará este requisito como aceptado si siempre que la aplicación se encuentre mostrando una escena de combate es posible volver al menú principal utilizando el botón dedicado para ello.

<b>ID</b>	RF-13
<b>Nombre</b>	Visualizar combate entre agentes
<b>Requisitos asociados</b>	<ul style="list-style-type: none"><li>▪ RF-6</li><li>▪ RF-10</li><li>▪ RF-11</li></ul>
<b>Descripción</b>	El sistema deberá permitir visualizar la totalidad de un combate entre el agente y otra versión de si mismo sin que el usuario tenga que controlar a ningún personaje dentro de la escena de combate.
<b>Precondición</b>	Se ejecuta la opción del menú dedicada a visualizar el combate entre dos agentes.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario pulsa el botón dedicado a comenzar la pelea entre dos agentes.</li></ol>
<b>Postcondición</b>	Se entra en la escena de combate con los dos personajes siendo controlados por un agente.
<b>Criticidad</b>	Vital
<b>Criterio de validación</b>	Se considerará este requisito como aceptado si se puede visualizar un combate completo entre un agente y otra versión de si mismo de la misma forma que se visualizaría un combate entre regular entre el usuario y un agente o entre dos usuarios.



<b>ID</b>	RF-14
<b>Nombre</b>	Simular múltiples combates
<b>Requisitos asociados</b>	<ul style="list-style-type: none"><li>■ RF-6</li></ul>
<b>Descripción</b>	El sistema deberá permitir que se realicen varios combates entre dos versiones del agente de forma acelerada para permitir que el mismo complete el proceso de aprendizaje en un tiempo razonable.
<b>Precondición</b>	Se ejecuta la opción del menú dedicada a simular varios combates entre dos agentes.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario pulsa el botón dedicado a simular varios combates entre agentes.</li></ol>
<b>Postcondición</b>	Se comienza la simulación de los combates.
<b>Criticidad</b>	Vital
<b>Criterio de validación</b>	Se considerará este requisito como aceptado si se pueden realizar varias simulaciones de forma iterativa en las que el agente visite los diferentes estados.

<b>ID</b>	RF-15
<b>Nombre</b>	Comandos en la consola
<b>Requisitos asociados</b>	<ul style="list-style-type: none"><li>■ RF-4</li></ul>
<b>Descripción</b>	El sistema deberá permitir ejecutar comandos utilizando la consola para realizar acciones específicas dentro de la aplicación de forma aislada.
<b>Precondición</b>	La consola de la aplicación está siendo visualizada.
<b>Secuencia normal</b>	<ol style="list-style-type: none"><li>1. El usuario introduce un comando en formato de texto en la consola.</li><li>2. El usuario pulsa el botón de ejecutar el comando.</li><li>3. El usuario visualiza el resultado.</li></ol>
<b>Excepciones secuencia normal</b>	<ol style="list-style-type: none"><li>4. El usuario no percibe ningún resultado porque el comando no existe.</li></ol>
<b>Postcondición</b>	El comando ha sido procesado y ejecutado.
<b>Criticidad</b>	Deseable
<b>Criterio de validación</b>	Se considerará este requisito como aceptado si se puede ejecutar un comando en la consola de la aplicación de forma que si el mismo es correcto se ejecute y si no lo es no ocurra nada.

**3.1.3.3. Requisitos no funcionales**

Los requisitos no funcionales especifican necesidades del sistema referentes a su operación, esto puede referirse a rendimiento, tiempos de respuesta, capacidad, etc. Al contrario que los requisitos funcionales, los requisitos no funcionales se centran en las características de funcionamiento de esas funcionalidades y no en las funcionalidades en si mismas.

En este apartado se definirán los requisitos no funcionales de este proyecto formalmente:

<b>ID</b>	RNF-1
<b>Nombre</b>	Rendimiento de la aplicación
<b>Descripción</b>	La aplicación, especialmente en las escenas de combate, deberá visualizarse a 60 fotogramas por segundo para garantizar la fluidez de cara al usuario.
<b>Criticidad</b>	Importante
<b>Criterio de validación</b>	Se considerará este requisito como cumplido si durante un número suficiente de combates consecutivos, los fotogramas por segundo medios de la aplicación no caen por debajo de 60.

<b>ID</b>	RNF-2
<b>Nombre</b>	Velocidad de las simulaciones
<b>Descripción</b>	En aras de permitir la realización del proceso de aprendizaje del agente en un tiempo razonable se deberá permitir la simulación de combates con el tiempo interno del juego sustancialmente acelerado.
<b>Criticidad</b>	Vital
<b>Criterio de validación</b>	Se considerará este requisito como cumplido si el agente es capaz de competir con una de las implementaciones basadas en reglas y ganar más de un 50 % de las veces después de menos de 15 minutos de entrenamiento.

<b>ID</b>	RNF-3
<b>Nombre</b>	
<b>Descripción</b>	
<b>Criticidad</b>	
<b>Criterio de validación</b>	

<b>ID</b>	RNF-4
<b>Nombre</b>	
<b>Descripción</b>	
<b>Criticidad</b>	
<b>Criterio de validación</b>	

<b>ID</b>	RNF-5
<b>Nombre</b>	
<b>Descripción</b>	
<b>Criticidad</b>	
<b>Criterio de validación</b>	

<b>ID</b>	RNF-6
<b>Nombre</b>	
<b>Descripción</b>	
<b>Criticidad</b>	
<b>Criterio de validación</b>	

<b>ID</b>	RNF-7
<b>Nombre</b>	
<b>Descripción</b>	
<b>Criticidad</b>	
<b>Criterio de validación</b>	

<b>ID</b>	RNF-8
<b>Nombre</b>	
<b>Descripción</b>	
<b>Criticidad</b>	
<b>Criterio de validación</b>	

#### 3.1.3.4. Matriz de trazabilidad

Como parte final de análisis de requisitos se muestra una matriz de trazabilidad en la Tabla 3.1 que hará corresponder a los requisitos funcionales definidos anteriormente con los casos de uso a los que están asociados

Cuadro 3.1: Matriz de trazabilidad de requisitos contra casos de uso

	CU-1	CU-2	CU-3	CU-4	CU-5	CU-6	CU-7	CU-8	CU-9
RF-1	✓		✓						
RF-2			✓						
RF-3			✓	✓					
RF-4			✓						✓
RF-5		✓							
RF-6			✓	✓			✓	✓	
RF-7							✓	✓	
RF-8							✓	✓	
RF-9							✓	✓	
RF-10			✓				✓	✓	✓
RF-11			✓				✓	✓	
RF-12			✓	✓					
RF-13				✓	✓				
RF-14				✓		✓			
RF-15			✓						

# Capítulo 4

## Gestión del proyecto

Dentro de la ingeniería de software, una de las partes esenciales para la realización de proyectos considerados exitosos es la gestión de proyectos. La realización de una buena gestión no se puede considerar, ni mucho menos, una garantía de que el proyecto a gestionar vaya a resultar un éxito. Sin embargo, si elegimos ignorar o realizar una mala gestión de nuestro proyecto sí podremos considerar que nos encontraremos en una situación proclive para un proyecto fallido. Durante toda la extensión del ciclo de vida de nuestro proyecto se utilizará la gestión de proyectos como un método que nos ayudará a lograr la obtención de un producto final ajustado a todas las necesidades y restricciones presentes, sea cual sea la índole de las mismas (tiempo, costes, requisitos, etc.).

Dedicaremos este capítulo a definir y explicar la gestión de nuestro proyecto en todas sus partes. Se determinará y explicará el análisis de riesgos, la metodología de desarrollo empleada, la gestión de configuración, la planificación temporal y la estimación de costes.

### 4.1. Gestión de riesgos

### 4.2. Gestión de la configuración

### 4.3. Metodología de desarrollo

Completar los riesgos necesarios

Completar la gestión de la configuración

Completar la metodología

## 4.4. Planificación temporal

Completar  
la plani-  
ficación  
temporal

## 4.5. Análisis de costes

Completar  
la esti-  
mación  
de costes

## 4.6. Plan de Gestión de las Comunicaciones

Completar  
el plan  
de  
Gestión  
de las  
Comuni-  
caciones



# Capítulo 5

## Arquitectura y herramientas

El IEEE define arquitectura como los conceptos fundamentales o propiedades de un sistema en su entorno que se encarnan en sus elementos, relaciones y los principios de su diseño y evolución [1]. Como referencia el propio IEEE, es el estándar ISO 12207 el que identifica un diseño arquitectónico y un diseño detallado y desgranado en lo que a un sistema se refiere. A partir de esto se obtiene que el diseño de la arquitectura de un sistema describe la estructura y la organización del mismo, es decir, se centra en los subsistemas o componentes que forman el sistema completo y las relaciones entre los mismos desde un punto de vista de alto nivel.

Contando ahora con una idea definida de cuales son los objetivos y requisitos de nuestro proyecto, nos centraremos en este capítulo en el diseño a alto nivel y en la arquitectura a la cual nos referimos en el párrafo anterior. Para ello utilizaremos representaciones gráficas a alto nivel del sistema y detallaremos los elementos arquitectónicos del producto y que tecnologías han sido usadas en los mismos así como las herramientas de las que hemos hecho uso para llevar a cabo el proyecto en su totalidad.

### 5.1. Arquitectura del sistema

### 5.2. Herramientas utilizadas


Completar  
Arquitect-  
tura del  
sistema

Completar  
Herra-  
amientas  
utiliza-  
das



# Capítulo 6

## Diseño e implementación



Completar  
Diseño  
e imple-  
menta-  
ción



# Capítulo 7

## Validación y pruebas

---

Completar  
Validación y  
pruebas



# Capítulo 8


## Valoraciones finales



Completar  
Valora-  
ciones  
finales







Quitar  
la sec-  
ción de  
ejemplos



# Capítulo 9

## Exemplos

### 9.1. Un exemplo de sección

Esta é *letra cursiva*, esta é **letra negrilla**, esta é letra subrallada, e esta é **letra curier**. Letra tiny, scriptsize, small, large, Large, LARGE e moitas más. Exemplo de fórmula:  $a = \int_0^\infty f(t)dt$ . E agora unha ecuación aparte:

$$S = \sum_{i=0}^{N-1} a_i^2. \quad (9.1)$$

As ecuaciones se poden referenciar: ecuación (9.1).

#### 9.1.1. Un exemplo de subsección

O texto vai aquí.

#### 9.1.2. Outro exemplo de subsección

O texto vai aquí.

##### 9.1.2.1. Un exemplo de subsubsección

O texto vai aquí.

##### 9.1.2.2. Un exemplo de subsubsección

O texto vai aquí.

##### 9.1.2.3. Un exemplo de subsubsección

O texto vai aquí.

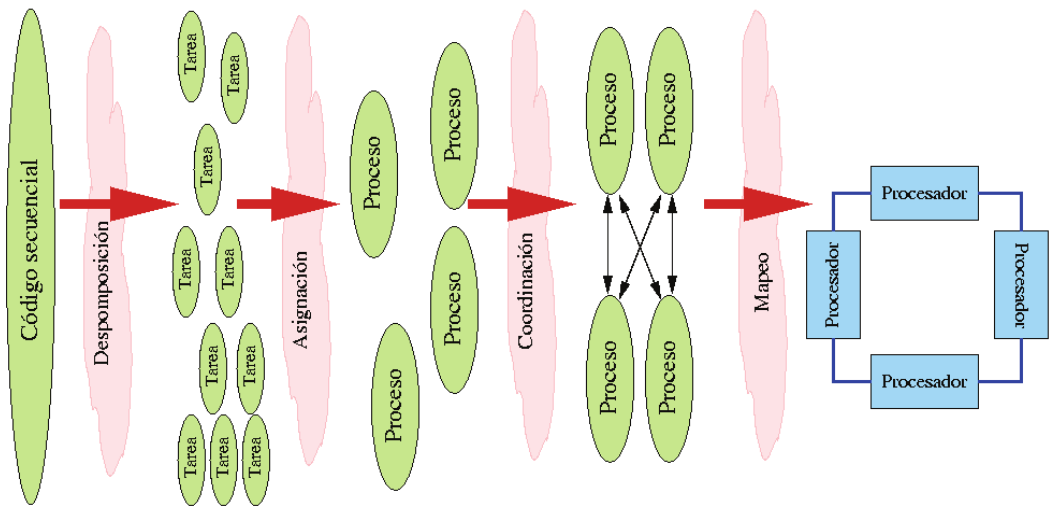


Figura 9.1: Esta é a figura de tal e cal.

Izquierda	Derecha	Centrado
ll	r	cccc
llll	rrr	c

Cuadro 9.1: Esta é a táboa de tal e cal.

9.2. Exemplos de figuras e cadros

A figura número 9.1.  
O cadro (taboa) número 9.1.

9.3. Exemplos de referencias á bibliografía

Este é un exemplo de referencia a un documento descargado da web [4]. E este é un exemplo de referencia a unha páxina da wikipedia [5]. Agora un libro [6] e agora unha referencia a un artigo dunha revista [7]. Tamén se poden por varias referencias á vez [4, 6].

9.4. Exemplos de enumeracións

Con puntos:

- Un.

- Dous.
- Tres.

Con números:

1. Catro.
2. Cinco.
3. Seis.

Exemplo de texto verbatim:

```
0 texto          verbatim
    se visualiza tal
        como se escribe
```

Exemplo de código C:

```
#include <math.h>
main()
{  int i, j, a[10];
   for(i=0;i<=10;i++) a[i]=i; // comentario 1
   if(a[1]==0) j=1; /* comentario 2 */
   else j=2;
}
```

Exemplo de código Java:

```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello_World!"); // Display the string.
    }
}
```



# Capítulo 10

## Conclusiones y posibles ampliaciones

---

Conclusiones e posibles ampliaciones

Completar  
conclu-  
siones y  
posibles  
amplia-  
ciones





# Apéndice A

## Manuais técnicos

Manuais técnicos: en función do tipo de Traballo e metodoloxía empregada, o contido poderase dividir en varios documentos. En todo caso, neles incluírase toda a información precisa para aquelas persoas que se vaian a encargar do desenvolvemento e/ou modificación do Sistema (por exemplo código fonte, recursos necesarios, operacións necesarias para modificacións e probas, posibles problemas, etc.). O código fonte poderase entregar en soporte informático en formatos PDF ou postscript.

Realizar  
manual  
técnico



## Apéndice B

### Manuais de usuario

Manuais de usuario: incluírán toda a información precisa para aquelas persoas que utilicen o Sistema: instalación, utilización, configuración, mensaxes de erro, etc. A documentación do usuario debe ser autocontida, é dicir, para o seu entendemento o usuario final non debe precisar da lectura de outro manual técnico.

Realizar  
manual  
de usua-  
rio



# Apéndice C

## Licencia

Insertar  
licencia  
apropia-  
da



Completar  
biblio-  
grafía





# Bibliografía

- [1] Definición de arquitectura según la ISO/IEC/IEEE 42010:2011. Artículo referenciado (<http://www.iso-architecture.org/ieee-1471/defining-architecture.html>). Consultado el 2 de junio de 2017.
- [2] Grady Booch, James Rumbaugh, and Ivar Jacobson. El lenguaje unificado de modelado. Manual de referencia. Addison Wesley, 1999.
- [3] IEEE recommended practice for software requirements specifications. Technical report, 1998.
- [4] Nvidia CUDA programming guide. Versión 2.0, 2010. Disponible en <http://www.nvidia.com>.
- [5] Acceso múltiple por división de código. Artigo da wikipedia (<http://es.wikipedia.org>). Consultado o 2 de xaneiro do 2010.
- [6] R.C. Gonzalez e R.E. Woods, *Digital image processing*, 3ª edición, Prentice Hall, New York, 2007.
- [7] P. González, J.C. Cartex e T.F. Pelas, “Parallel computation of wavelet transforms using the lifting scheme”, *Journal of Supercomputing*, vol. 18, no. 4, pp. 141-152, junio 2001.



Quitar  
esta lista  
de TO-  
DOs



# Todo list

Completar la organización del documento . . . . .	3
Completar requisitos asociados y sus nombres . . . . .	14
Completar los riesgos necesarios . . . . .	35
Completar la gestión de la configuración . . . . .	35
Completar la metodología . . . . .	35
Completar la planificación temporal . . . . .	36
Completar la estimación de costes . . . . .	36
Completar el plan de Gestión de las Comunicaciones . . . . .	36
Completar Arquitectura del sistema . . . . .	37
Completar Herramientas utilizadas . . . . .	37
Completar Diseño e implementación . . . . .	39
Completar Validación y pruebas . . . . .	41
Completar Valoraciones finales . . . . .	43
Quitar la sección de ejemplos . . . . .	45
Completar conclusiones y posibles ampliaciones . . . . .	51
Realizar manual técnico . . . . .	53
Realizar manual de usuario . . . . .	55
Insertar licencia apropiada . . . . .	57
Completar bibliografía . . . . .	59
Quitar esta lista de TODOs . . . . .	63

