CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

TRABAJO FIN DE GRADO

Edición Semántica Inteligente de Entornos Virtuales

Autor: Adrián Calle Murillo

Director: Angélica de Antonio Jiménez

MADRID, JUNIO 2015

Adrián Calle Murillo – Escuela Técnica Superior de Ingenieros Informáticos

AGRADECIMIENTOS

En estas breves líneas, me gustaría agradecer a todo el que ha colaborado para que este trabajo de Fin de Grado sea posible, ya sea directa o indirectamente.

Primeramente dar las gracias a Angélica, por su dedicación y sobradas capacidades intelectuales en este trabajo como tutora y en todo lo que conlleva el proyecto LORO, el cual seguro que con ella estará en buenas manos. Sin duda ha sido una buena elección trabajar contigo, y lo volvería a hacer sin duda.

En segundo lugar a mis compañeros del laboratorio: Álvaro, Bego y David Calero. Teneros al lado trabajando es un placer chicos, y este trabajo no hubiera sido posible sin vuestra ayuda, vuestros ánimos y los buenos momentos vividos juntos en todas las horas de trabajo que hemos compartido en el "labo".

También a mi compañero David Álvarez. Compañero de mis andanzas en esta carrera desde que la empecé y desde antes de empezarla. Grandísimo programador y mejor persona; tus consejos, ayudas y alguna que otra bronca me han servido para ser mejor profesional de este sector y crecer. Sin ti no habría llegado hasta aquí para escribir estas líneas.

También no olvidarme a Pablo y Sergio. Amigos en la carrera y fuera de ella.

A mi familia. Mi madre, mi hermano, mis abuelos,... Pero sobre todo a mi padre, José Antonio, el que desde que tengo uso de razón se empeñó en que fuera alguien en esta vida y tratando de que fuera una persona versada y con estudios. Este trabajo tiene mucho de ti, una parte importantísima. Por fin. Ahora sí que puedo decirlo. Los años de esfuerzo han merecido la pena. Lo he conseguido, papá.

Y por último a Laya y a Nuria, las que me han enseñado todo lo que no se enseña en las universidades. La vida en definitiva. Por vuestra inagotable energía, que al final ha calado en mí, y por ser quienes sois y existir en definitiva. Ha sido una gran suerte coincidir con vosotras. Soy así gracias a vosotras y este trabajo también es vuestro.

Por último, me gustaría agradecer (y dar mis disculpas) a todos aquellos que, de una u otra forma, han contribuido al desarrollo de este trabajo, y que he olvidado mencionar.

RESUMEN

Tradicionalmente, los entornos virtuales se han relacionado o vinculado de forma muy estrecha con campos como el diseño de escenarios tridimensionales o los videojuegos; dejando poco margen a poder pensar en sus aplicaciones en otros ámbitos. Sin embargo, estas tendencias pueden cambiar en tanto se demuestre que las aplicaciones y ventajas de estas facilidades software, se pueden extrapolar a su uso en el ámbito de la enseñanza y el aprendizaje. Estas aplicaciones son los conocidos como Entornos Virtuales Inteligentes (EVI); los cuales, tratan de usar un entorno virtual para llevar a cabo labores de enseñanza y tutoría, aportando ventajas como simulación de entornos peligrosos o tutorización personalizada; cosa que no podemos encontrar en la mayoría de los casos de las situaciones de enseñanza reales.

Este trabajo trata de dar solución a una de las problemáticas que se plantean a la hora de trabajar con cualquier entorno virtual con el que nos encontremos y prepararlo para su cometido, sobre todo en aquellos enfocados a la enseñanza: dotar de forma automática e inteligente de una semántica propia a cada uno de los objetos que se encuentran en un entorno virtual y almacenar esta información para su posterior consulta o uso para otras tareas. Esto quiere decir que el objetivo principal de este trabajo, es el proceso de recolección de información que se considera importante de los objetos de los entornos virtuales, como pueden ser sus aspectos de la forma, tamaño o color. Aspectos que, por otra parte, son realmente importantes para poder caracterizar los objetos y hacerlos únicos en un entorno virtual donde, a priori, todos los objetos son los mismos a ojos de un ordenador.

Este trabajo que puede parecer trivial en un principio, no lo es tanto; y servirá de sustento fundamental para que otras aplicaciones futuras o ya existentes puedan realizar sus tareas. Una de estas tareas pudiera ser la generación de indicaciones en lenguaje natural para guiar a usuarios a localizar objetos en un entorno virtual, como es el caso del proyecto LORO sobre el que se engloba este trabajo. Algunos ejemplos de uso de esta tarea pueden ser desde ayudar a cualquier usuario a encontrar sus llaves en su propia casa a ayudar a un cirujano a localizar cierta herramienta en un quirófano. Para ello, es indispensable conocer la semántica e información relevante de cada uno de los objetos que se presentan en la escena y diferenciarlos claramente del resto.

La solución propuesta se trata de una completa aplicación integrada en el motor de videojuegos y escenarios 3D de mayor soporte del mundo como es Unity 3D, el cual se interrelaciona con ontologías para poder guardar la información de los objetos de cada escena. Esto hace que la aplicación tenga una potencial difusión, gracias a las herramientas antes mencionadas para su desarrollo y a que está pensada para tanto el usuario experto como el usuario común.

ABSTRACT

Traditionally, virtual environments have been related to tridimensional design and videogames; leaving a little margin to think about its applications in other fields. However, this tendencies can be changed as soon as it is proven that the applications and advantages of this software can be taken to the learning and teaching environment. This applications are known as intelligent virtual environments, these use the virtual environment to perform teaching and tutoring tasks; tasks we cannot find in most real life teaching situations.

This project aims to give a solution to one of the problematics that appears when someone works with any virtual environments we may encounter and prepare it for its duty, mainly those environments dedicated to teaching: automatically and intelligently give its own semantic to the objects that are in any virtual environment and save this information for its posterior query or use in other tasks. The main purpose of this project is the information recollection process that considers the different important facts about the objects that are in the virtual environments, such as their shape, size or color. Facts that are very important for characterizing the objects; to make them unique in the environment where the objects are all the same to the computer's eye.

This project may seem banal in the beginning, but it is not, it will be the fundamental base for future applications. One of this applications may be a natural language indicator generator for guiding users to locate objects in a virtual environment, such as the LORO project, where this project is included. Some examples of the use of this task are: helping any user to find the keys of his house, helping a surgeon to find a tool in an operation room… For this goals, it is very important to know the semantics and the relevant information of each object of the scenario and differentiate each one of them from the rest.

The solution for this proposal is a fully integrated application in the videogame and Unity 3D engine that is related to ontologies so it can save the object's information in every scenario. The previously mentioned tools, as well as the idea that this application is made for an expert user as well as for a common user, make the application more spreadable.

ÍNDICE

ÍNDICE DE FIGURAS

CAPÍTULO 1- INTRODUCCIÓN

1.1. INTRODUCCIÓN

La generación de indicaciones para la localización de objetos en entornos y situaciones de la vida real puede parecernos a priori algo trivial y sencillo de realizar; sin embargo, cuando trasladamos este problema a los entornos virtuales, se convierte en un problema interesante y a la vez complejo de solucionar. Los entornos virtuales no son inteligentes de por sí como podría ser un humano dando indicaciones en el mundo real con total facilidad; por ello, es interesante dotar de esta inteligencia a los entornos virtuales y equipararlo a la inteligencia del ser humano para hacer las indicaciones de la forma más precisa y clara posible.

El **proyecto LORO** nace en el Laboratorio Decoroso Crespo de la Escuela Técnica Superior de Ingenieros Informáticos (Universidad Politécnica de Madrid) como parte de esta idea y formando parte desde hace años de una tesis doctoral en curso dirigida por la doctora Angélica de Antonio Jiménez. Trabajando sobre la idea de **entornos virtuales inteligentes** (EVI) preparados para la tutorización y la enseñanza de usuarios, el proyecto consiste en la generación de indicaciones en lenguaje natural para ayudar al usuario a la localización de objetos en entornos virtuales tridimensionales. Todo ello de forma adaptativa en función del características del usuario y del entorno virtual; es decir, totalmente personalizado y como un sistema que pueda aprender por sí mismo para ofrecer la mejor experiencia y casi tan real como si de un ser humano se tratara dándonos las indicaciones. Para la generación de las indicaciones se emplean los parámetros de saliencia (nivel de vistosidad de algo de forma cuantitativa a la percepción visual) individual de los objetos y de contexto; además del conocimiento del objeto por parte del usuario (ya sea previo o no), sus capacidades cognitivas y perceptivas y su comprensión de áreas de conocimiento del mundo. Todo este se encuentra definido en un complejo modelo de saliencia total para cada objeto y su entorno para dar las indicaciones, pero esto no forma parte de este trabajo.

El proyecto se divide en dos partes bien diferenciadas: el proceso de pre procesamiento de los entornos virtuales, con el objetivo de preparar los datos necesarios de **saliencia individual** (color, tamaño y forma) de los objetos y otros muchos necesarios que se recogen de cada uno y guardarlos en ontologías mediante un editor semántico; y el proceso de ejecución, el cual recoge los datos de pre procesamiento y detalla una indicación elaborada para un objeto a localizar en base a objetos de referencia el entorno. Estos objetos de referencia serán tomados basándonos en la información semántica del pre procesamiento y los datos del usuario y entorno. Adicionalmente, el proyecto integra la gestión de usuarios del mismo para el uso de ellos en el sistema.

Este **Trabajo Fin de Grado** radica en la primera parte de este proyecto y la integración con la segunda parte ya mencionada antes junto con la gestión de usuarios del sistema.

El trabajo ha consistido en detallar un modelo de la saliencia individual de los objetos y crear una aplicación en Unity 3D para guardar de forma automática los datos necesarios para este modelo de saliencia (y otros como el tamaño, orientación, material…) en ontologías; este es el conocido como editor-exportador del proyecto LORO. Además, el modelo de saliencia individual ha sido testeado sobre usuarios reales para verificar que los resultados que muestra sobre los objetos que resaltan más en un entorno virtual, se adapta a la percepción de los usuarios; y por tanto, es válido como modelo. Esto se ha realizado mediante el experimento de la métrica de la forma y basándose en varios artículos con experimentos ya realizados, que se referenciarán más adelante.

Las ontologías desarrolladas en el trabajo se han basado en un modelo de usuarios-mundo, que se detallará con posterioridad, con el objetivo de separar ambos y poder distinguirlos en su manejo y almacenamiento, ya que, el editor-exportador desarrollado en este trabajo se encarga de gestionar las áreas de conocimiento del mundo y la información de los objetos; y por otra parte, la interfaz de integración a modo de vista de *login* para los usuarios, se encarga de gestionar los usuarios, su percepción del mundo, sus capacidades y acceder a las escenas para empezar a solicitar indicaciones acerca de los objetos del entorno.

Todos los detalles de estos aspectos que se acaban de mencionar, se realizarán en la parte de desarrollo de este trabajo del segundo capítulo de esta memoria y en las conclusiones del tercer capítulo.

El desarrollo de este trabajo permitirá el dotar de semántica a los objetos de cualquier escenario tridimensional virtual con el que nos encontramos; ya que, recoge gran variedad de propiedades y datos de los objetos para su posterior procesado. Esto implica que la aplicación desarrollada puede ser usada en otros aspectos de los EVI o los entornos virtuales en general y aprovechar las ventajas que el editor semántico proporciona. Con esto, se solucionaría uno de los principales problemas de los entornos virtuales para su recolección automática de datos de los objetos y exportarlo a los modelos de conocimiento ontológicos que tanto auge en el campo de la investigación están teniendo.

Por último, mencionar que este trabajo es la continuación de un trabajo previo de Prácticum por parte del autor y de muchos trabajos previos de varios alumnos que colaboraron en el proyecto LORO; y que por tanto, hay trabajo ya avanzado y que están en otros TFG o Prácticum. Sin embargo, muchas de las tareas desarrolladas en el trabajo son totalmente empezadas desde el inicio y sin referencias anteriores para su realización. Esto se detallará con claridad en el documento, para distinguir los aspectos que han sido tratados en este trabajo como parte de desarrollo del mismo.

1.2. ESTRUCTURA DE LA MEMORIA

Esta memoria se trata de un documento donde se reflejan los avances y trabajos desarrollados por el autor durante su Trabajo de Fin de Grado como Graduado en Ingeniería Informática. La memoria se puede estructurar en varias partes bien diferenciadas para poder facilitar su lectura y seguimiento.

Primeramente, nos encontramos con los agradecimientos y el resumen en español e inglés que dan paso a la memoria en sí de este trabajo.

El **primer capítulo**, en el que se encuentra precisamente este apartado, resalta una introducción sobre el tema del que trata este trabajo, los objetivos planteados del mismo y una breve reseña de los trabajo previos que se han realizado sobre el tema y que han podido ser un apoyo al autor para hacer este trabajo.

El **segundo capítulo** contiene el grueso de esta memoria, con todo lo que implica el desarrollo del trabajo realizado por parte del autor: herramientas empleadas, metodología de trabajo, marco conceptual, pruebas del sistema, manual de usuario y el desarrollo de todo el trabajo en base a lo anterior.

El **tercer capítulo** versa sobre las conclusiones referidas a lo que se ha expuesto previamente en el segundo capítulo de la memoria del trabajo; así como futuras líneas de trabajo para el proyecto LORO o en otros aspectos, y que por razones u otras no se han completado en este trabajo o no eran de su responsabilidad.

Adicionalmente, nos encontramos con anexos y referencias bibliográficas; los cuales, se encuentran debidamente referenciados en todo el documento.

1.3. ESTADO DE LA CUESTIÓN

El proyecto LORO lleva desde el año 2012 en marcha. Y desde la incorporación del autor a este proyecto durante septiembre del año 2014 son muchos los alumnos que han participado activamente en el desarrollo del proyecto y de la tesis doctoral en la que se engloba. Estos alumnos son Diego Dotor, Roberto Mendoza y Javier Gimeno.

Primeramente, Diego Dotor definió un algoritmo cuyo objetivo primordial es localizar el llamado objeto de referencia (RO) en un entorno virtual, y sobre el cual será el mejor candidato para dar las indicaciones del objeto a localizar (LO), en base a su saliencia individual con respecto a todos los objetos del entorno y la distancia al observador o usuario. El algoritmo recoge las "frecuencias" de color, tamaño y forma de cada uno de los objetos, para posteriormente ordenarlos por el más saliente, sumar cada una de las frecuencias de cada objeto con pesos y dar el RO correcto. Estas "frecuencias" se basan en un modelo de saliencia individual muy primitivo que desarrolló Roberto Mendoza y que detalla más abajo. A esto se añade el que se tiene en cuenta una holgura de distancia de los objetos a la vista del observador; con lo cual, los objetos que se encuentren a menos de un 10% de distancia del observador y el tamaño del escenario, serán válidos para ordenarlos según su saliencia individual y candidatos a ser el RO de este escenario.

También Diego comenzó a indagar sobre el uso de ontologías en .NET y C# para guardar los datos de las saliencias de los objetos y dejó la idea de usar pesos diferentes para cada valor de la saliencia individual, detalle fundamental para el posterior desarrollo de los objetivos de este trabajo y del modelo de saliencia.

A su vez, Roberto se aproxima a desarrollar el modelo de saliencia individual de cada objeto y definir el cómo reflejar y calcular cada una de sus tres componentes básicas: color, tamaño y forma. Para el color aproxima la idea de dar el color más saliente según el entorno virtual y sus colores; es decir, pasando el color a CieLab compara el color del objeto con el color promedio del escenario. Si coinciden la saliencia será 0, sino se promediará en función a su "lejanía" a este color promedio gracias a las facilidades que ofrece el trabajar en un espacio CieLab de color. Para el aspecto de tamaño, no se definió ninguna forma que con certeza se dé un valor promediado. Sin embargo, para la forma se definieron y se usaron métricas muy acertadas para su determinación. Para ello, se emplean los descriptores de Zernike (una serie de vectores exclusivos de cada objeto que tratan de describir su forma unívocamente) y el cálculo de volúmenes con los voxeles (como pequeños cubos unitarios que conformen la forma objeto) de un objeto.

Para finalizar, Javier se encargó de varias partes muy importantes y que son la base de este trabajo en su gran medida. Javier aproximó la definición de las ontologías empleadas en el proyecto LORO, que en sus posteriores y mejoradas versiones se han desarrollado en este trabajo; además de que, empleó el sistema de comunicación de ontologías que ya

comenzó Diego con Jena sobre IKVM, y con éxito. Sin embargo, estas ontologías resultaron estar mal diseñadas y no suponían una gran ayuda para el desarrollo del exportador de este proyecto. También determinó el cálculo de los parámetros de contexto para la generación de indicaciones como la visión periférica del objeto o el tiempo de visión y recuerdo del objeto, y las características completas de los usuarios como pueden ser la probabilidad de recuerdo o el conocimiento previo de los objetos y áreas de conocimiento del sujeto. Esto permitió a Javier hacer la aproximación más exacta hasta entonces del comienzo de este trabajo en un proyecto sobre Unity 3D, sobre el objetivo de generar indicaciones en base a las características individuales de los objetos, las características del usuario y el entono. Obviamente, ha sido mejorado considerablemente.

Bien es cierto, que estos trabajos previos sentaron los modelos teóricos más aproximados y básicos de lo que será el resultado del modelo de saliencia individual de este trabajo y del propio exportador que llevará los datos automáticamente a las ontologías creadas en este trabajo. Pero los resultados de este trabajo, distan bastante de los resultados de estos trabajos mencionados; aunque su base claro está, son todos ellos.

Destacar en último lugar, que el autor en trabajos anteriores de Prácticum en el proyecto LORO consiguió modelar una pequeña parte el modelo de saliencia, el cual ha sido perfeccionado y terminado en este trabajo; además de diseñar las primeras versiones del editor-exportador y de las ontologías. Estas versiones han sido tomadas para ser también perfeccionadas en este trabajo, ya que como las versiones de Javier, han resultado ser versiones mal diseñadas.

Por ello, en este trabajo se detallan las últimas versiones del modelo de saliencia, editor-exportador y ontologías definitivas para el proyecto LORO y que han supuesto la cumbre de los meses de trabajo en este proyecto por parte del autor.

1.4. OBJETIVOS PLANTEADOS

Una vez detallados los trabajos previos del proyecto sobre el que se embarca este trabajo y de haber resaltado de manera introductoria los propósitos del mismo, se van ahora a desglosar de forma breve y concisa los objetivos que se plantean en este trabajo:

- Mejorar las versiones preliminares del editor-exportador y ontologías realizados durante el periodo de Prácticum por parte del autor.
- Mejorar el modelo de saliencia individual de los objetos que se encuentran en las escenas de entorno virtuales tridimensionales.
- Realizar experimentos para la evaluación de las métricas de saliencia individual, que son tales como tamaño, color y forma.
- Integrar el sistema del editor y la recolección de datos de los objetos con el sistema de ejecución del proyecto LORO mediante una interfaz de *login* de usuarios del sistema.
- Mejorar la usabilidad del editor-exportador a nivel de usuario.
- Encontrar nuevas formas de manejar ontologías en .NET.
- Diseñar y ejecutar un plan de pruebas del sistema del editor completo.
- Llevar a cabo la evaluación del uso del sistema por diseñadores de entornos virtuales.
- Elaborar el manual de uso.
- Documentar el diseño realizado.
- Planificar y realizar el seguimiento del proyecto utilizando técnicas y herramientas de gestión de proyectos.
- Realizar tareas de investigación para la comprensión de las herramientas empleadas y para ampliar el modelo de saliencia individual.
- Integrar el sistema con periféricos de realidad virtual y aumentada.
- Dotar de un sistema de clasificación automática de objetos por forma y predecir la clase seleccionada de la ontología en el editor-exportador.
- Realizar pruebas con expertos en usabilidad para el editor-exportador.

Destacar que estos objetivos son de gran amplitud de tareas, y que cada uno de ellos implica muchas horas de trabajo y la realización de sub-objetivos para poder llegar hasta su completitud.

CAPÍTULO 2 – DESARROLLO

2.1. MARCO CONCEPTUAL

El trabajo se engloba en muchas concepciones teóricas que han sido desarrolladas durante los años de desarrollo del proyecto LORO o en este mismo trabajo. Como ya se ha dicho durante el primer capítulo, se tratará de reseñar cuál de estos han sido parte de realización del autor o de los trabajos previos que han sido detallados anteriormente.

A grandes rasgos, el marco conceptual se puede dividir en cuatro partes bien diferenciadas, cada una a su vez con sus sub partes y marco conceptual propio. Estas partes son: el modelo de saliencia individual, el manejo de ontologías en el modelo usuarios-mundo, los entornos virtuales inteligentes (EVI) y el modelo de usuarios y su conocimiento.

2.1.1. MODELO DE SALIENCIA INDIVIDUAL

Este modelo de saliencia individual se corresponde con la primera parte del modelo de saliencia total del que se compone el proyecto LORO en el que se engloba el trabajo de esta memoria. Después esta saliencia se usará para poder calcular la saliencia total en base a la saliencia de contexto (ver FIGURA 1).

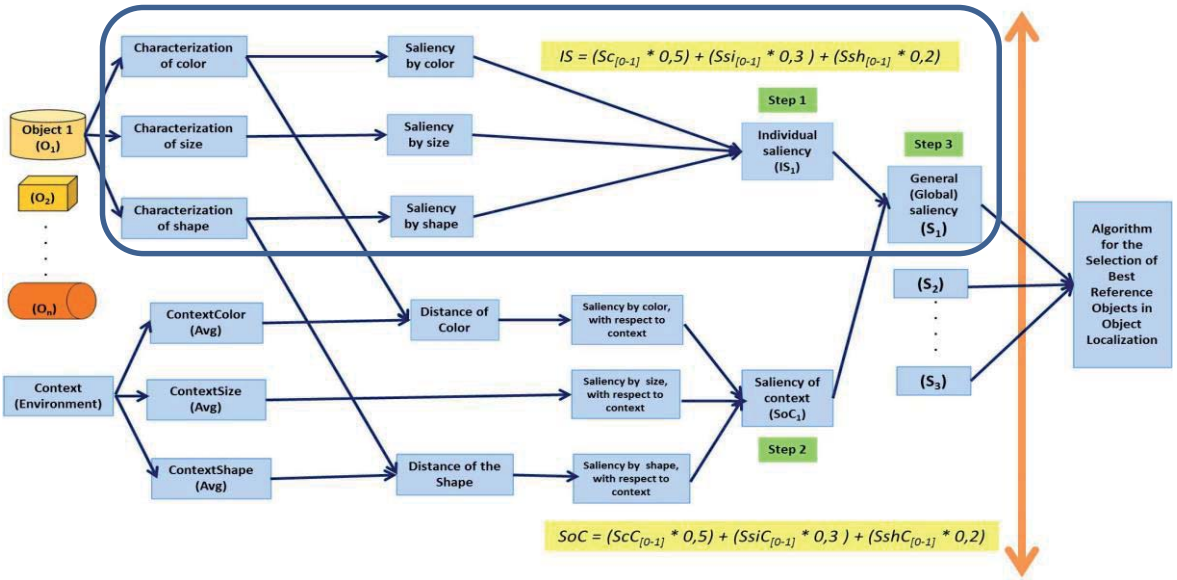


FIGURA 1. Esquema de saliencias del proyecto LORO. Mostrando el que versa este trabajo rodeado en azul.

La saliencia individual que vamos a contar en este apartado, se calcula de forma íntegra por el editor-exportador, en tiempo de pre compilación, que es el objetivo principal de este trabajo y que se detallará más adelante. Por su parte, la saliencia de contexto se calcula durante la ejecución de la aplicación del proyecto LORO para la generación de indicaciones en lenguaje natural. Esto es debido, a que como sus propios nombres indican, la saliencia por contexto determina el nivel de vistosidad a la percepción visual conforme a parámetros del entorno como pueden ser el resto de objetos o la percepción del usuario y sus conocimientos; mientras que la saliencia individual es algo totalmente diferente.

La **saliencia individual** es un término que hace referencia al nivel de vistosidad o de atracción a la percepción visual que un objeto por sí mismo y sin ser influenciado por nada más, es capaz de producir en un ser humano.

Encontrar un modelo matemático preciso para poder determinar la saliencia individual es algo que no es nada trivial. Basándose en varias referencias de autores que han aproximado una solución a este problema (véase Ref. 1, 2 y 3), se ha tratado de crear un modelo matemático preciso. De las referencias antes citadas se han extraído ideas tanto para la saliencia individual que nos atañe como la de contexto. Por ejemplo el autor *Gapp* en su artículo ya hace referencia a usar parámetros del tamaño y del contexto para poder determinar un orden de saliencia para cada uno de los objetos que nos encontramos en un entorno virtual (véase Ref. 1). Los autores del artículo "*A Model of Saliency-Based Visual Attention for Rapid Scence Analysis*" (véase Ref. 3) van mucho más allá y consiguen determinar una fórmula matemática para la saliencia visual de los objetos de una imagen basándose en los términos que ellos indican de "color" predominante, "intensidad" de contraste de luz y "orientación" en grados de 0 a 135.

El modelo matemático que se ha propuesto para este trabajo en concreto, toma las ideas como ya se ha dicho de estos artículos, se ha desarrollado durante las prácticas del autor en el Prácticum y en este trabajo; siempre tratando de mejorar los resultados del modelo matemático y tratando que se asemejen lo más posible a la realidad de cómo un usuario percibiría los objetos más salientes. Decir también que este modelo de saliencia se encuentra mucho más detallado en la tesis doctoral en curso sobre la que se apoya este trabajo y el proyecto LORO como ya se ha dicho (véase Ref. 18), y que el modelo de saliencia individual es fruto completamente de este trabajo, del proyecto LORO y esta tesis.

Este modelo de saliencia basa su fórmula matemática en emplear tres parámetros que también poseen cada uno una forma diferente de ser calculados: saliencia por color, saliencia por forma y saliencia por tamaño.

2.1.1.1. Saliencia por color

Los colores en los objetos son algo que es lo primero que la capacidad visual de un usuario percibe. Pero determinar cuáles son los colores que pueden resaltar más en el cerebro de un usuario es una tarea que no es fácil de estandarizar y determinar. Sin embargo, para el proyecto LORO se han acogido las ideas reflejadas por el autor A. Choungourian (véase Ref. 4) para poder determinar esta difícil tarea. Este autor ya propone cuales son los colores más vistosos o "preferidos" para los usuarios en función de su cultura y condición (ver FIGURA 2). Con ello el modelo de saliencia que nosotros proponemos se basa en estos colores, para los cuales el trabajo ha sido determinar un compendio de estos colores más vistoso por culturas para hacerlos uno solo (ver FIGURA 3).

ORDER AND PERCENTAGE OF EACH COLOR PREFERRED OVER THE REST OF THE COLORS BY FOUR NATIONAL GROUPS

Order	U.S.A.	Lebanon	Iran	Kuwait
1st	Red** 66.1%	Green** 66.1%	Blue-green** 68.2%	Blue-green** 67.5%
2nd	Blue** 65.7%	Red 55.0%	Green* 56.8%	Yellow-green** 60.4%
3rd	Green** 57.8%	Yellow-green 54.3%	Red 50.0%	Green** 57.8%
4th	Orange 48.6%	Blue 50.4%	Blue 48.9%	Orange* 56.8%
5th	Yellow 45.0%	Blue-green 49.6%	Orange 48.2%	Purple 47.1%
6th	Yellow-green† 43.2%	Orange 46.1%	Yellow-green 44.6%	Yellow‡ 38.9%
7th	Purple‡ 39.6%	Yellow‡ 40.7%	Yellow† 43.6%	Red‡ 37.1%
8th	Blue-green‡ 33.9%	Purple‡ 37.8%	Purple‡ 39.6%	Blue‡ 34.3%

‡ or ** Critical Ratio value (z) significant at .01 level.
† or * Critical Ratio value (z) significant at .05 level.

FIGURA 2. Colores preferidos por grupos nacionales y culturales (ver Ref. 4).

Color	Código RGB	Código CieLab
Rojo	255,0,0	53.23,80.1,67.22
Naranja	255,112,40	64.16,51.02,62.62
Amarillo	255,255,0	97.13,-22.55,94.88
Amarillo Verdoso	154,205,50	76.53,-37.49,66.58
Verde	0,255,0	87.63,-60.18,83.18
Azul Verdoso	13,152,186	58.11,-20.81,-26.94
Azul	0,0,255	32.30,79.19,-107.86
Morado	102,2,153	27.20,59.25,-56.42

FIGURA 3. Compendio de colores más salientes basados en el artículo de Choungourian, con sus colores en RGB y CieLab.

Como se puede observar, los colores más salientes se muestran en código CieLab a parte de RGB. Esto tiene una razón de ser por dos motivos.

Primeramente, porque el uso de un espacio de color CieLab permite el calcular distancias entre colores de una forma mucho más precisa que con un espacio de RGB y que se mucho más eficiente su uso. Esto está demostrado en el artículo "*A study of efficiency and accuracy in the transformation from RGB to CIELAB color space*" (véase Ref. 6) donde se habla de cómo pasar de un color RGB a uno CieLab y las ventajas del uso de este espacio de color. La transformación de RGB a CieLab que será empleado en la aplicación del editor-exportador, ya que el motor de Unity 3D sólo trabaja con RGB (véase Ref. 16), se basa en el modelo matemático pasando por XYZ que da este artículo.

La segunda razón del uso de CieLab es porque nuestros potenciales usuarios pueden padecer dificultades visuales para poder percibir los colores de los objetos. Estos tipos de daltonismo se tienen en cuenta para poder "adaptar" la saliencia por color en base a lo que el usuario se espera que vea del objeto. Esto se detallará en el modelo de usuarios de este capítulo. La transformación de los colores RGB a RGB para daltónicos se ha extraído del artículo "Intelligent Modification of Colors in Digitized Paintings for Enhancing the Visual Perception of Color-blind Viewers" (véase Ref. 5) y se basa en un modelo matemático de matrices para poder realizar la correcta conversión.

Destacar que estos dos algoritmos referenciados con anterioridad, el de conversión a CieLab en RGB y de cambio a colores daltónicos según su tipo de RGB a RGB, han sido implementados por el alumno en código C# para que pueda usarlo Unity 3D. Algunos de estos algoritmos ya estaban diseñados por alumnos en trabajos previos (véase Pág. 14); sin embargo, estos códigos no pasaron las pruebas realizadas en la primera versión del editor-exportador que ese verá más adelante (véase Pág. 55). Por ello, el autor de este trabajo decidió implementarlos desde cero para poder obtener los resultados esperados en el trabajo y en el proyecto LORO.

Conocidos ya estos detalles, la obtención de la métrica de la saliencia por es bastante sencilla. Dependiendo del color en CieLab que hayamos extraído del objeto y convertido al daltonismo correspondiente si fuera necesario, simplemente se trata de coger la tabla de los colores más salientes que se ve arriba (ver FIGURA 4) y si los tres valores LAB caen en alguno de los colores en sus correspondientes LAB con una holgura de más menos 5 unidades; entonces, se le asigna un valor 1 de saliencia por color a este objeto. De lo contrario, el valor asignado al objeto para la saliencia por color sería de 0. Siempre un valor asignado de entre estos dos y basado en la tabla que hacemos generado del compendio de los colores más preferidos de Choungourian.

2.1.1.2. Saliencia por tamaño

Los tamaños de los objetos son algo que también destaca a la hora de determinar cuánto de llamativos son; obviamente, un monitor de ordenador es más llamativo que el ratón del propio ordenador, simplemente porque es más grande que él, por ejemplo.

Por ello, la métrica de la saliencia por tamaño pudo ser la más trivial de pensar de cómo dotarla de una métrica concreta. Este modelo es completamente propio del proyecto LORO y se basa en una idea muy simple. Empleando los parámetros del alto, ancho, largo y volumen de un objeto; ponderamos cada uno de ellos de 0 a 1 con todos los objetos que ya se encuentran exportados en la ontología donde se guarda la información, donde 1 es el más alto, largo, ancho o voluminoso, y el resto se ponderan sobre este valor. La ponderación sobre el valor se realiza dividiendo el valor del objeto a ponderar entre el del objeto que sea la referencia de la ponderación por tener el valor más alto de toda la ontología en referencia a uno de estos valores (alto, largo, ancho y volumen). Obviamente, diferentes objetos pueden ser la referencia de ponderación en cada uno de los valores mencionados anteriormente. Y posteriormente, se realiza la media aritmética de estos cuatro valores para volver a obtener un valor entre 0 y 1 (ver FIGURA 4).

$$Saliencia\ Tamaño = \frac{(\text{Pond. Alto} + \text{Pond. Largo} + \text{Pond. Ancho} + \text{Pond. Volumen})}{4}$$

FIGURA 4. Fórmula del modelo de saliencia por formal. Proyecto LORO.

Los valores de largo, ancho y alto son valores que se obtienen fácilmente gracias al motor de física del que dispone Unity 3D y que no da estos valores en centímetros sin ningún problema (véase Ref. 16). El problema viene con determinar el volumen de los objetos.

Aquí entran en juego los procesos de **voxelización** de los objetos.

El proceso de voxelización de los objetos es uno de los mayores problemas de este trabajo. Consiste en tratar de determinar el volumen de los objetos y su forma mediante el uso de voxeles de tamaño unitario y definido. Estos voxeles se encargan de recorrer el cuerpo del objeto completo y de llenarlo con estos voxeles hasta completar su volumen total.

Luego, el volumen se puede calcular con el número de voxeles y cogiendo el tamaño de ambos en centímetros cúbicos para multiplicarlo por cada uno de los voxeles. Y así es como se ha hecho para obtener el volumen de cada uno de los objetos.

Pero sin embargo, debido a razones esenciales este algoritmo se ha vuelto un quebradero de cabeza como se ha dicho. Originalmente, el algoritmo de voxelización fue un diseño de varios alumnos que trabajaron antes en el proyecto LORO. Este algoritmo

lanzaba pares de rayos en las tres dimensiones del objeto y de forma paralela para poder determinar donde se encuentra el comienzo y fin de un tramo del objeto, por así decirlo; y después calcular el número de voxeles que entrarían. Este algoritmo el problema que tiene es su pobre eficiencia, ya que tarda muchísimo tiempo y se basa en fuerza bruta y lanzar rayos a todo el escenario hasta dar con el objeto. En pruebas estimadas en este trabajo, para poder voxelizar objetos nativos de los assets de Unity 3D tarda de un orden de segundos dependiendo del tamaño del objeto y hasta del orden de minutos (en incluso el sistema se queda colgado en su ejecución) cuando se trata de objetos importados de otros programas de modelado como pueden ser Maya o 3DS. Este hecho es algo inadmisible para una aplicación de estas características como el editor-exportador y para nuestro modelo de saliencia.

Por ello en este trabajo, el autor optó por alternativas a nuevos algoritmos de voxelización. El resultado de horas de trabajo trajo consigo el algoritmo que actualmente se encuentra implementado: *Unity Voxelization Master* (véase Ref. 9). Este algoritmo es mucho más eficiente y trata el proceso de voxelización de una forma diferente. Se basa en emplear pequeños triángulos que determinan la mitad de un voxel; pero éstos no se buscan con rayos como se hacía antes, sino que los triángulos se generan en la malla de los objetos. Así se ganan dos ventajas: primero porque se sabe dónde buscar en un principio y un final al comparar pares de triángulo predefinidos en la malla y se rellena los huecos entre ellos con voxeles; y además, los triángulos son más precisos para poder detectar las singularidades de las formas al ser más adaptada que un cubo de voxel del algoritmo anterior (ver FIGURA 5). El resultado es voxelizaciones realizadas en tiempo de décimas de segundo tanto en objetos nativos de Unity 3D como en los exportados de otras aplicaciones (ver FIGURA 6).

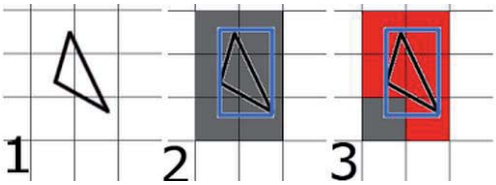


FIGURA 5. Uso de triángulos para el proceso de voxelización (Véase Ref. 9).

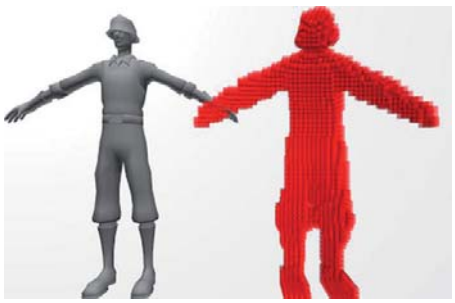


FIGURA 6. Resultado del proceso de voxelización de un objeto OBJ (Véase Ref. 9).

2.1.1.3. Saliencia por forma

A pesar de que puede parecer que la saliencia por forma es algo que se solucione fácilmente conociendo las dificultades encontradas en el apartado anterior, es algo que también ha supuesto problemas para determinar cómo obtener su saliencia para cada uno de los objetos del entorno.

En este caso había tres alternativas: emplear el ratio del espacio vacío, usar los descriptores de Zernike o la clasificación entrenada de formas descrita en "Shape Classification Through Structured Learning of Matching Measures" (véase Ref. 8). En nuestro caso para el proyecto LORO, se ha empleado el ratio de espacio vacío para la saliencia individual y los descriptores para la saliencia de contexto.

La última opción del artículo de **clasificación con aprendizaje estructurado por emparejamientos de medidas**, fue descartada con anterioridad a la incorporación del autor de esta memoria al proyecto LORO. Pero se trata de la base o idea sobre la que se apoya el ratio de espacio vacío.

Los **descriptores de Zernike** son una serie de vectores que se encargan de analizar la estructura de los objetos en formato OBJ y determinar con ellos las formas del mismo. Estos descriptores se extraen de un programa externo al editor-exportador y ha sido materia de investigación fuera de este trabajo (véase Ref. 7); es decir, se limitado el autor de esta memoria a arrancar este programa y recoger los datos que genera en formato de texto para llevarlos al editor-exportador y completar la información de los objetos. Pero no es algo clave para este trabajo.

Aunque no se hayan usado los descriptores de Zernike en este modelo de saliencia, si suponen datos a recoger por parte del exportador para poder facilitárselos a los siguientes modelos de saliencia en la segunda parte del procesamiento del proyecto LORO.

El **ratio de espacio vacío** es la métrica empleada para poder determinar la "llamatividad" de un objeto por su forma basándose en lo que pueda destacar este en sus huecos vacíos o lo picudo que pueda ser. Esta métrica simplemente la relación entre el espacio vacío en centímetros cúbicos del objeto respecto a su *Bounding Box* de Unity 3D y el espacio total del mismo *Bounding Box* (ver FIGURA 7).

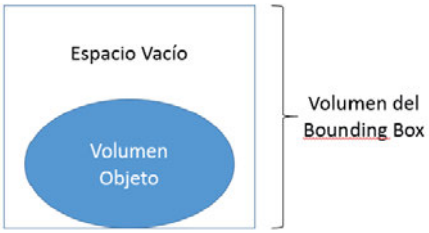


FIGURA 7. Idea del ratio de espacio vacío. Proyecto LORO.

Un *Bounding Box* es la "caja" mínima que contiene un objeto en sus tres dimensiones. Con esta métrica se puede entonces decir que aquellos cuyo valor sea mayor es que son más salientes (tienen más huecos y "picudeces") y los que menos salientes sean son los que llenan casi por completo el *Bounding Box*. Para poder normalizar la métrica simplemente tomamos el más saliente por esta métrica como 1 de entre todos los que están en la ontología y el resto se pondera en base a este valor, dividiendo su métrica entre la del más saliente. Con la misma idea de las ponderaciones del tamaño, obtenemos valores entre 0 y 1 para esta métrica.

Una vez se hayan obtenido todas estas variables, que repetimos se encuentran normalizadas cada una entre 0 y 1, se pasa a calcular la **totalidad de la saliencia individual** de cada objeto mediante la fórmula matemática que da sentido a todo este modelo. Como se ve más abajo (ver FIGURA 8), el modelo de saliencia individual se encuentra ponderado con una serie de pesos para cada una de las tres variables, dando más importancia a la saliencia por color, después por forma y por último el tamaño.

Saliencia Individual = (0.5 * Saliencia Color) + (0.3 * Saliencia Forma) + (0.2 * Saliencia Tamaño)

FIGURA 8. Fórmula del modelo de saliencia individual. Proyecto LORO.

Con todo esto ya tendríamos las saliencia por forma de cada uno de los objetos que se meterán en la ontología correspondiente y se emplearán para posteriores partes del proyecto LORO como son su integración a la saliencia total para encontrar el objeto de referencia mejor para poder dar las indicaciones al usuario. Como también se puede observar, los valores de la saliencia individual también se encuentran entre 0 y 1.

2.1.2. MARCO ONTOLÓGICO. ONTOLOGÍA MODELO USUARIOS-MUNDO.

Para poder llevar a cabo las labores del editor-exportador y almacenar toda la información que se extrae del modelo de saliencia individual, el modelo de usuarios (véase Pág. 31) o la información relevante para el proyecto LORO; se necesita una forma de almacenar este conocimiento, y éste va a ser usar los sistemas inteligentes de las ontologías.

Una **ontología** es una forma de representar el conocimiento del mundo, una forma de almacenar información. Puede parecernos lo mismo que pudiera ser una base de datos, pero no es así. Las ontologías son capaces de inferir datos y axiomas gracias a su sistema de relaciones entre las instancias en los que se basa por el uso del lenguaje RDF de la W3C. Otras de las ventajas de las ontologías es que su uso se puede reusar en nuevas otología que quieran usar recoger un conocimiento parecido, son totalmente reutilizables y de una buena comprensión.

El proyecto LORO al completo, necesita del uso de ontologías para poder gestionar todo el conocimiento que engloba. Por lo tanto, su comprensión y uso por parte del autor ha sido un conocimiento nuevo a adquirir y clave para el desarrollo de las tareas y alcanzar los objetivos del trabajo.

Las ontologías se componen de tres elementos en el enfoque de uso de lenguaje RDF (Resource Description Framework), lenguaje para poder definir recursos que las componen:

- Clases → Se trata de la abstracción de elementos del conocimiento que queremos representar. Un ejemplo es "Padres" o "Hijos", en el conocimiento de una familia reflejado en una ontología.

- Propiedades → Se trata de las relaciones o predicados que se establecen entre las clases (propiedades objeto) o para ellas mismas (propiedades de datos). Un ejemplo de propiedad objeto puede ser "esHijoDe" y una propiedad de dato podría ser "Edad". Referido siempre entre clases de una ontología como hemos dicho.

- Instancias → Se trata de la especificación de estas abstracciones en casos concretos del mundo que conocemos tanto para las clases como para las instancias, y representan el conocimiento que queremos almacenar. Se representan con tripletas RDF. Un ejemplo de instancia de clase Padre puede ser Juan.

En las ontologías sólo pueden existir este tipo de recursos presentados antes para poder trabajar con ellas y diseñarlas. Aunque como ya se ha destacado, es posible hacer muchas más cosas gracias a su sistema de inferencia y de razonamiento.

En resumen, las clases y las propiedades son como el *schema* de las bases de datos relacionales y las instancias son los datos que metemos en cada una de las tablas de estas bases de datos que todos conocemos (ver FIGURA 9).

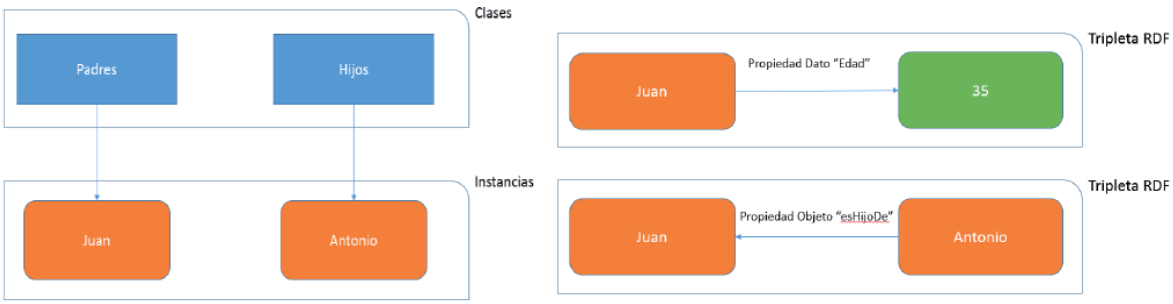


FIGURA 9. Ejemplo de representación de elementos de las ontologías en lenguaje RDF.

El diseño de las ontologías para el proyecto LORO por parte de este trabajo, lleva sufriendo cambios constantes desde la incorporación del autor al proyecto. En unas primeras versiones, y como se ha destacado en los trabajos previos a este, el colaborador del proyecto en el laboratorio Decoroso Crespo Javier Gimeno, realizó unos primeros diseños de las ontologías con tres ontologías interconectadas entre sí: mundo, usuarios y cosas.

Desde la incorporación del autor el diseño se realizó de nuevo para adaptarlo a las necesidades del proyecto y del editor-exportador. Así que durante el período de Prácticum del autor se realizó un nuevo diseño con dos ontologías y siguiendo el modelo que había propuesto Javier: mundo y usuarios. Metiendo dentro de mundo la ontología de cosas para simplificar las ontologías.

Pero cuando se empezó este trabajo por parte del autor, surgió un problema muy grave que había que solucionar. Y es que las ontologías que se encuentran interconectadas o importadas unas con otras, como es el caso de las dos versiones de antes mencionadas, provocan una ineficiencia del sistema del editor-exportador y la interfaz de integración brutales; debido a que Jena (ver en herramientas usadas Pág. 34) al cargar las ontologías en memoria, tarda mucho en realizar las tareas que le piden estos sistemas como pueden ser el cargar o guardar datos en las ontologías. Esto lastra la experiencia del usuario y del experto que usa el editor-exportador.

Por ello el autor decidió en este trabajo unificar los modelos de las ontologías en uno solo, basándose en el modelo de usuario-mundo y el diseño recomendado por varios autores de la realización de ontologías para su mayor eficiencia y comprensión (véase Ref. 10, 11 y 14).

Las ontologías del proyecto LORO son una para cada escena que integre esta aplicación y se derivan de una ontología modelo que contiene el esqueleto (clases y propiedades) sobre las que se van a crear las demás ontologías que llamaremos las ontologías específicas.

Esta **ontología modelo** se basa en el **modelo de usuario-mundo**. Esta es una forma de diseñar ontologías para representar el conocimiento que implique a usuarios de ese conocimiento y el conocimiento en sí; como nos ocurre en nuestro trabajo y en el proyecto LORO.

La forma de abordar este diseño se encuentra reflejado en una serie de artículos de varios autores para realizarlo (véase Ref. 10, 11 y 14). Pero básicamente, el modelo se basa en dejar en una misma o en diferentes (como ya se intentó en versiones preliminares) separados los usuarios del conocimiento del mundo. Así se consigue una mayor comprensión del conocimiento y una mayor eficiencia por tener separado que le corresponde a cada usuario del mundo.

En la ontología modelo ya diseñada (ver diagrama de clases y propiedades Anexo A) en la que almacenamos información de varias fuentes y todas relativas a información para cualquier escenario por parte del editor-exportador o del proyecto LORO:

- Objetos → Se guardan instancias de todos los objetos del mundo con sus propiedades como la saliencia individual, su propietario o su material del que está hecho.
- Usuarios → Se almacenan instancias de los usuarios que "pertenecen" a esta ontología y por tanto a un escenario. Ver apartado del modelo de usuarios de este mismo capítulo para más información (véase Pág. 31).
- Áreas de conocimiento → Se guardan las instancias de las áreas de conocimiento que van con este escenario para relacionarlas con los objetos mediante la propiedad de "belongsTo" o a los usuarios (ver apartado de Modelo de usuarios Pág. 31).
- Objetos conocidos por el usuario → Almacenamos la información relativa a los objetos que conoce el usuario. Ver apartado del modelo de usuarios de este mismo capítulo para más información (véase Pág. 31).
- Áreas conocidas por el usuario → Se almacenan en las ontologías para saber la comprensión o cuán es de experto un usuario en cierta área de conocimiento. Ver apartado del modelo de usuarios de este mismo capítulo para más información (véase Pág. 31).

2.1.3. ENTORNOS VIRTUALES INTELIGENTES (EVI'S).

Bien es cierto que todo lo que ya se ha contado (y lo que está por venir) está muy bien, pero cómo pretendemos que el usuario interactúe con estos modelos de saliencia, que seleccione los objetos que quiera localizar en el proyecto LORO, que el editor-exportador exporte qué objetos de qué escenario y con qué características y para qué tipo de escenario en concreto,…

Todas estas respuestas están en los **Entornos Virtuales Inteligentes (EVI)**. Se trata de entornos visuales diseñados especialmente para realizar tareas de simulaciones y tutorías inteligentes con el objetivo de que los usuarios aprendan a hacer ciertas tareas o cometidos. Las ventajas que proporcionan estos escenarios son múltiples con respecto a las tutorías o enseñanzas que se pueden hacer en la vida real y en esas condiciones.

Pero veámoslo con un ejemplo real de EVI para que todo esto que se está detallando se pueda entender perfectamente.

En el laboratorio Decoroso Crespo, donde se lleva a cabo este trabajo y el proyecto LORO, se emplean las EVI para llevar a cabo sus objetivos de todos sus proyectos. Uno de ellos es el proyecto PIPE comenzado el año 2010. El objetivo de este proyecto es simular varios laboratorios de investigación reales, uno de ellos puede ser el de biotecnología (ver FIGURA 10), con el objetivo de que alumnos realicen prácticas en este entorno virtual.

FIGURA 10. Ejemplo de EVI para las tutorías en un laboratorio de biotecnología. Proyecto PIVE del Laboratorio Decoroso Crespo.

Las ventajas que proporcionan estos entornos virtuales inteligentes en este caso son variadas, por ejemplo: poder realizar prácticas de importante riesgo en el mundo de la ingeniería genética sin tener que poner en peligro a los alumnos y sin el gasto que supone el tener material caro para su realización, tener siempre disponible un tutor que aprende de tus errores y te guía de forma inteligente para que alcances tus objetivos, simular situaciones de paso del tiempo,…

Gracias a este ejemplo se ha comprobado que los entornos virtuales pueden ser importantísimos para hacer tutorías inteligentes y pasar de un ámbito de investigación a algo que se pueda emplear en cualquier etapa de las enseñanzas de una persona.

En el caso de este trabajo, se engloba en una EVI (ver FIGURA 11) para la generación de indicaciones en lenguaje natural para la localización de objetos en el proyecto LORO; y como ya se ha dicho antes, se encarga de preparar la escena con el editor-exportador con toda la información relativa a los objetos como su saliencia individual entre otras, meterlo en ontologías para su posterior uso y el hacer una interfaz de integración que haga de puente entre la parte de preparación y la de ejecución propiamente dichas.

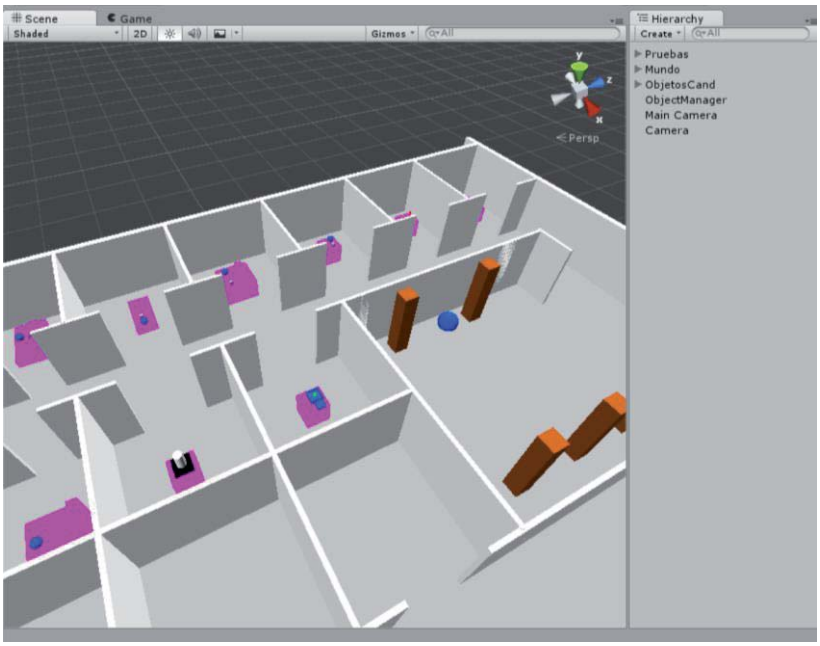


FIGURA 11. EVI preliminar del proyecto LORO para la generación de indicaciones. Como se puede observar, existen varias "salas" con diferentes situaciones para dar indicaciones al usuario.

2.1.4. MODELO DE USUARIOS.

Los usuarios son una parte muy importante en el proyecto LORO, ya que todo el modelo de saliencia por contexto depende en gran parte de ellos y lo que perciben del entorno en tiempo de uso de la aplicación.

Se puede pensar que esta parte no tiene nada que ver con lo que desarrolla este trabajo, pero no es así el todo. Este trabajo se ha encargado también de realizar el manejo de los usuarios en las ontologías junto con todos sus datos y características, como ya se ha visto en el apartado anterior del modelo de las ontologías, mediante la interfaz de integración del proyecto LORO que sirve a modo de ventana de *login* para que los usuarios se puedan crear, borrar, editar y ser seleccionados para trabajar con la aplicación del proyecto. Esto se verá más adelante.

Aquí se va a detallar el modelo de los usuarios que manejan las ontologías junto con sus características y peculiaridades.

Un **usuario** en el proyecto LORO es único en cada ontología, y por tanto en cada escenario, y en cada una de ellas puede tener características y conocimiento totalmente diferente al que pueda tener en otro entorno virtual. Esta decisión se tomó en base a que la importancia de los escenarios es muy alta y que da igual qué usuario que emplee el sistema, ya que los algoritmos de saliencias están preparados para ser inteligentes y dar las mejores indicaciones siendo cual sea el usuario del sistema potencial.

Las características de un usuario y que se reflejan en las características de la clase *Users* de las ontologías son las siguientes:

- Nivel de daltonismo → Se contempla como una de las características más importantes, ya que como se ha visto en el capítulo de la saliencia individual, dependiendo del nivel de daltonismo del usuario, se adaptan las indicaciones y por supuesto las saliencias de los objetos a los colores que realmente percibe de ellos. Existen y se contemplan en este trabajo **cuatro tipos de daltonismo**: ninguno (persona sana), tritanopia (carencia de sensibilidad al color azul, no se percibe), deuteranopia (carencia de sensibilidad al color verde, no se percibe) y la protanopia (carencia de sensibilidad al color rojo, no se percibe). Esto se refleja tanto en el perfil del usuario de la interfaz de login como en las ontologías como una propiedad de los usuarios. Y se emplea en la traducción de los colores a daltónico para adaptar la escena al usuario.

- Nombre de usuario y email → Datos de identificación de los usuarios, como en todo sistema de gestión de usuarios. Editable en el *login* y almacenado en las ontologías.

- Interpretar mapas → Propiedad que mide la capacidad del usuario de coger un mapa y saber "moverse" por él. Se almacena con los valores "Alto", "Medio" y "Bajo" en las ontologías. Editable desde el *login*.

- Capacidad de memoria → Propiedad que mide la capacidad del usuario de tener retentiva de ideas y sobre todo de formas de objetos, clave para hacer indicaciones más precisas en el sistema del proyecto LORO. Se almacena con los valores "Alto", "Medio" y "Bajo" en las ontologías. Editable desde el *login*.

- Capacidad de localizar objetos → Propiedad que mide la capacidad del usuario de intuir la localización de objetos que recuerda de su ubicación anterior. Es lo que llamaríamos el "¿Dónde he puesto…?" en la idea de que el usuario lo recuerde más o menos. Se almacena con los valores "Alto", "Medio" y "Bajo" en las ontologías. Editable desde el *login*.

- Agudeza visual → Propiedad guardada en las ontologías y editable en el *login* que es la capacidad de un usuario para percibir, detectar o identificar objetos especiales con unas condiciones de iluminación buenas gracias a su sistema de visión. Medido con el test de Snellen (ver FIGURA 12).

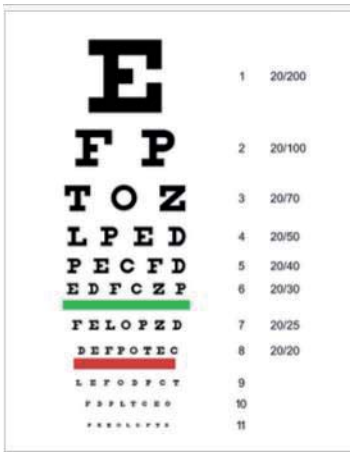


FIGURA 12. Gráfica del Test de Snellen para medir la agudeza visual de un usuario potencial del sistema del proyecto LORO.

- Comprensión de las áreas de conocimiento → En las ontologías se almacenan las áreas de conocimiento y los usuarios, como ya hemos visto. Esta propiedad relaciona las áreas que un usuario conoce en un escenario y almacena su nivel de comprensión de la misma con los valores "Alto", "Medio" y "Bajo". El objetivo es que si un objeto del entorno que pueda servir de objeto de referencia para las indicaciones, pertenece a un área de conocimiento en la que el usuario sea un experto, pues usaremos ese antes que otro de un área que desconozca o sea menos versado en ella. Esto se refleja en la ontología en tres instancias relacionadas con una relación de objeto "knowField" y

"isLinkedAreaWith" de la ontología para el usuario con el conocimiento de esa área del usuario y la propia área de conocimiento en sí.

Por ejemplo: "Juan" conoce el área "Juan_Informática" con un nivel "Medio" y este a su vez está conectado con el área de "Informática" de la ontología.

- Conocimiento de los objetos del entorno → Se trata de un caso muy parecido al del conocimiento del usuario en las áreas de las ontologías. En este caso, esta propiedad relaciona los objetos que un usuario conoce de un escenario y almacena su nivel de comprensión de la misma con los valores en varios parámetros.

Los **parámetros del conocimiento de un objeto** por parte de un usuario son los siguientes: tiempo de visión central, tiempo de visión periférica, claridad de visión (nivel de oclusión que poseía el objeto cunado se vio), última vez visto en visón central, última vez visto en visión parcial y superficie vista (cantidad de objeto respecto al usuario en superficie que fue última vez visto).

Como se puede observar, el usuario posee dos tipos de visión: central y periférica; las cuales se reflejan con el ángulo de visión donde el usuario centra su atención (visión central) y aquella donde ve pero no presta atención (visión periférica). Esto es importante guardarlo en las ontologías, ya que supone una gran forma de saber cómo se vieron los objetos y saber las condiciones de su conocimiento.

Al igual que las áreas, si de un objeto el usuario tiene buen recuerdo del objeto, éste será mejor candidato para poder ejercer las labores de objeto de referencia. Todo esto se almacena en las ontologías tanto en el proceso de ejecución como en el editor-exportador; ya que como se aprecia, son datos que se actualizan cada vez que el usuario entra en un escenario a ver objetos.

En las ontologías se refleja con tres instancias, al igual que las áreas, pero en este caso tenemos el usuario, el conocimiento de un objeto y el objeto en sí unidos por relaciones de objeto de las ontologías con "knowObject" y "isLinkedObjectWith".

Por ejemplo: "Juan" conoce el objeto "Juan_Silla01" con sus características pertinentes (las que aparecen más arriba en este sub apartado) y este a su vez está conectado con el objeto de "Silla01" que también está en las ontologías y tiene los datos de ese objeto como la saliencia individual y otros parámetros.

2.2. HERRAMIENTAS UTILIZADAS

Las herramientas que se han usado para la realización de este proyecto son las que más podían adaptarse a las necesidades de este. Siempre usando herramientas gratuitas y con gran soporte a nivel mundial para la ayuda del autor a realizar su trabajo, muchas de ellas se interrelacionan entre sí y son clave en el funcionamiento de otras.

Protegé es una herramienta gratuita y de código libre para editar ontologías y crear sistemas inteligentes basados en modelos de conocimiento ontológicos que ha sido desarrollada en la universidad de Stanford en Estados Unidos (véase Ref. 13). Posee una completa interfaz gráfica muy usable que permite realizar todo tipo de operaciones en la ingeniería ontológica y soportando todos los lenguajes del estándar de la W3C (ver FIGURA 13). Además tiene un gran soporte liderado por expertos en la materia, que está en constantes actualizaciones y con una gran comunidad de usuarios por detrás; lo que lo hace llegar a ser el editor de ontologías más usado y conocido en el mundo a nivel de investigación.

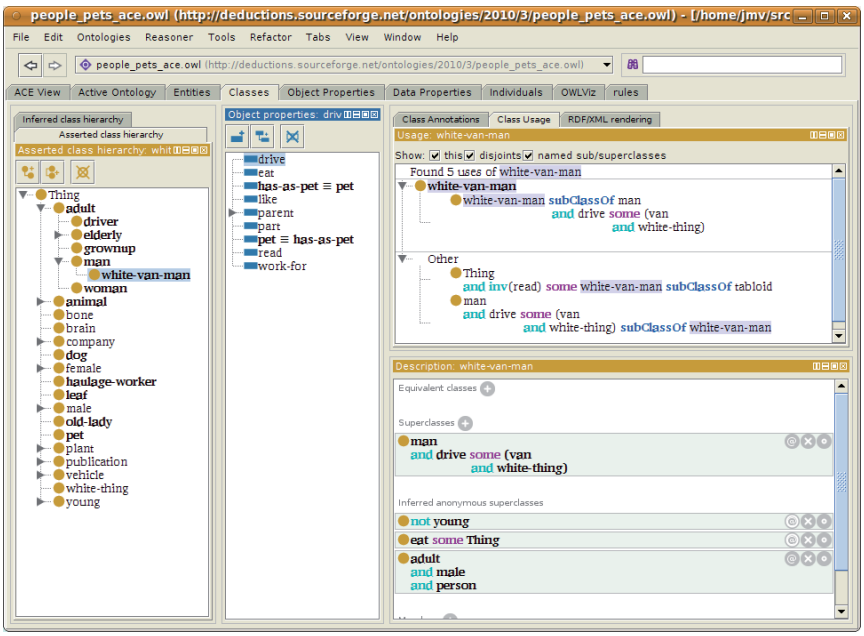


FIGURA 13. Ventana del software Protege.

Su uso en este trabajo es clave para poder diseñar las ontologías que ya se han detallado en anteriores apartados; y además para poder realizar las pruebas del editor-exportador en tanto lo que veamos que lo que se está llevando a las ontologías gracias a este exportador de la información de los objetos, es lo correcto y que se guarda tan y como se desea en un principio en el diseño de las ontologías.

Jena es un framework de Apache para trabajar sobre modelos (generalmente RDF) de ontologías desde código en Java, almacenando los modelos en memoria como si estuviéramos con las ontologías trabajando sobre Protegé (véase Ref. 17). Existe un *JavaDoc* de este *framework* para poder guiarse y usarlo correctamente, el cual se encuentra disponible en Internet. Se trata del framework más popular y completo para trabajar en modelos ontológicos, pudiendo realizar casi cualquier tarea que se nos plantee con las ontologías.

Para poder usar este *framework* en Unity 3D en este trabajo, es necesario el uso del compilador de Jena a C# mediante la generación de archivos DLL de .NET llamado **IKVM** (véase Ref. 15). Su objetivo es usar sus facilidades para usar el código C# como se haría en código Java, de forma totalmente transparente a la percepción del programador. Esto ya se había hecho antes en el laboratorio y fue un compañero del mismo el que ayudó al autor en la introducción de este tema y su uso.

Las facilidades de IKVM son básicamente que permite correr código compilado en Java (como el que usa Jena para gestionar las ontologías) directamente en MonoDevelop de C#. Mostramos ahora cómo se hace este proceso para que el autor haya podido trabajar en C# con Jena sin problemas (ver FIGURA 14).

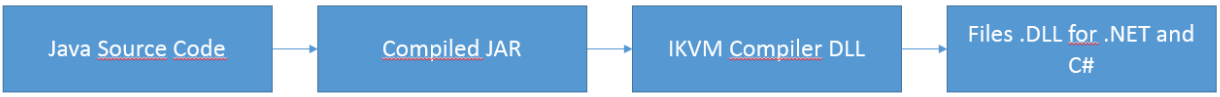


FIGURA 14. Diagrama de uso de IKVM para compilar código Java y correrlo en .NET.

Como vemos, estos DLL que genera IKVM se llevan al proyecto de Unity, que es .NET y C#; y entonces, se puede usar sin problemas Jena como si de código Java en vez de C# se tratara simplemente referenciando en el código C# estos DLL.

Owl Dot Net Api (ODNA) es un framework/API que sirve para manejar ontologías OWL/RDF en .NET lo que sería también válido para su uso con C# en Unity 3D. Es muy parecido al que antes se ha explicado sólo que este ofrece una visión más robusta, ya que no ejerce un paso por el código Java como hace IKVM; sino que va directamente a C#, ya que está diseñado solamente para este lenguaje y su framework.

Sin embargo, a una semana de la entrega de esta memoria, el autor destaca que la fuente de la que extrajo la librería del framework en cuestión, no se encuentra disponible en internet como hace unos meses. Por eso no se encuentra referenciada en la bibliografía; y que igualmente, IKVM y Jena pueden hacer sus mismas funciones en el trabajo en cuestión si ningún tipo de problema. Es decir, se pueden usar tanto Jena como ODNA, y como ODNA no está disponible en estos momentos; la recomendación del autor es usar Jena que está más extendido y tiene mejor soporte.

Oculus Rift es un casco de realidad virtual el cual actualmente se encuentra en versión de desarrollo, pero del cual se espera que haya una versión de consumidor para el año 2016. Famoso por sus aplicaciones en los videojuegos principalmente, también se puede emplear en los EVI para poder mejorar la experiencia de usuario y hacer más interactivos los mismos.

Se compone de una pantalla que hace efecto de 3D estereoscópico mediante dos lentes (una para cada ojo) y de varios giroscopios, acelerómetros y magnetómetros. Además, posee un kit de desarrollo para poder integrar su software en casi cualquier plataforma, como Unity 3D que es la que nos interesa, ya que permite que el proyecto LORO en su proceso de ejecución pueda verse de una forma totalmente novedosa para que el usuario reciba las indicaciones sobre los objetos que quiere localizar.

Unity 3D es el motor de videojuego más famoso y de mayor soporte en el mundo (véase Ref. 16). Su fama se debe a que para los creadores de videojuegos amateur posee una versión gratuita que cualquiera puede instalarse y usar en su ordenador; a pesar de que posee una versión de pago con muchas más utilidades. Esto facilita su acceso a cualquier usuario para poder introducirse en el mundo del desarrollo de videojuegos si tener que ser un verdadero profesional del sector. Posee una completa interfaz gráfica en una aplicación para poder usar sus facilidades (ver FIGURA 15).

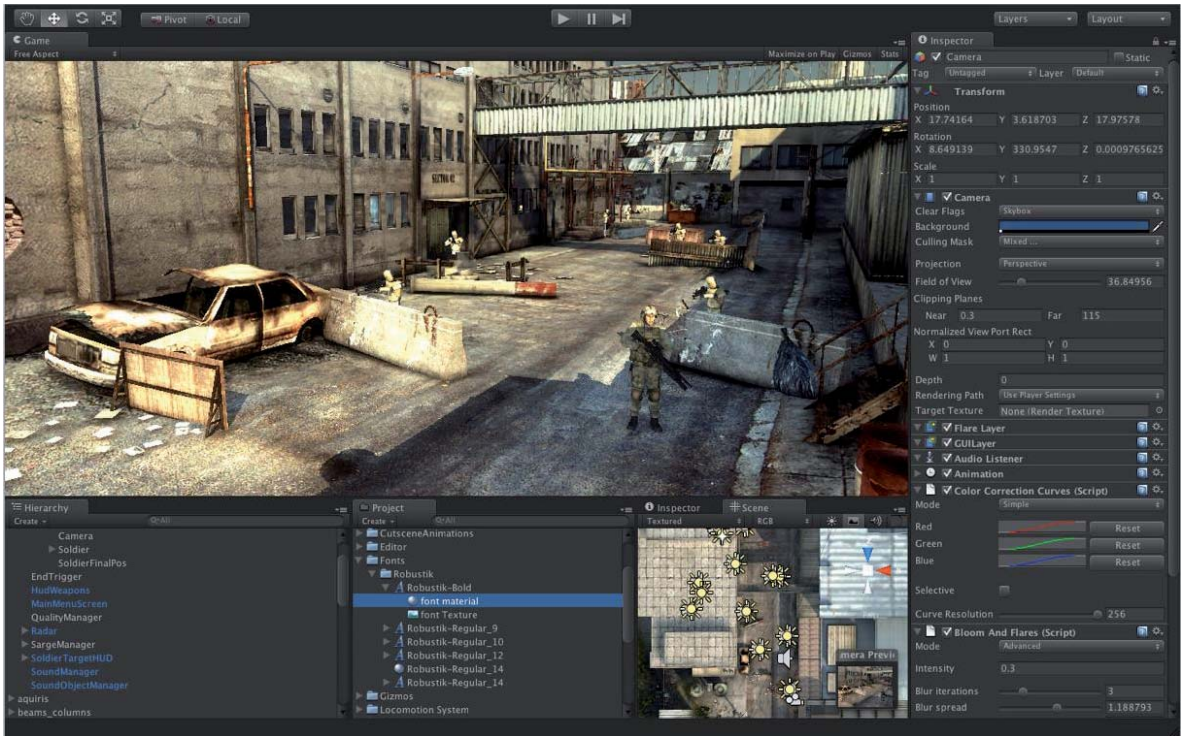


FIGURA 15. Ventana de la aplicación Unity 3D.

Este motor de videojuegos se puede emplear en aplicaciones más allá de los propios sistemas de entretenimiento digital. En este caso para este trabajo, se ha empleado para crear la aplicación del editor-exportador y la gran mayoría de las otras tareas que se han realizado en este trabajo. Unity 3D permite tanto diseñar escenarios, hacer scripts para el comportamiento de los componentes de los mismos (ver FIGURA 16), crear tu propio editor en la misma plataforma y aplicaciones con las facilidades del entorno (ver FIGURA 17). Todas estas facilidades hacen que sea sencillo llevar a cabo los objetivos planteados en este trabajo, ya que todas las aplicaciones, código y escenarios que han sido desarrollados están sobre esta plataforma (a excepción única de las ontologías). Unity 3D desarrolla sus scripts en C# y JavaScript, siendo el primero de los dos el que se ha empleado para el desarrollo de todo el código debido a su proximidad léxica y semántica con el resto de lenguajes de programación que se ven en la carrera.

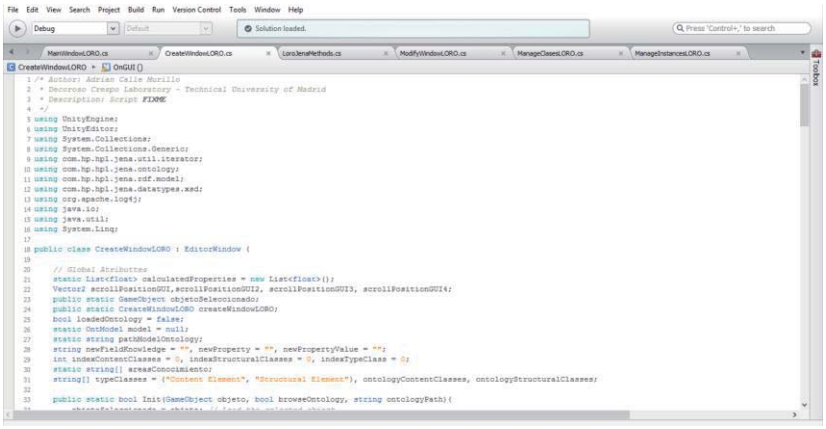


FIGURA 16. Entorno MonoDevelop de Unity 3D para desarrollar scripts.

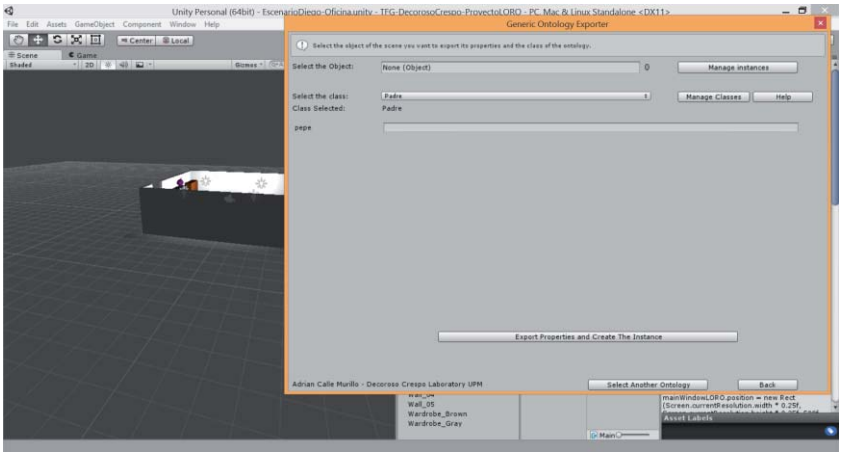


FIGURA 17. Ventana personalizada creada en el editor de Unity 3D. Versión primer exportador del proyecto LORO.

2.3. METODOLOGÍA DE TRABAJO Y PLANIFICACIÓN

La forma de llevar a cabo este trabajo y por ende esta memoria, ha sido mediante el trabajo diario en el Laboratorio Decoroso Crespo. Con un puesto asignado desde hace meses para el alumno en el propio laboratorio y el trabajo diario de 25 horas semanales durante 16 semanas, se ha conseguido completar con éxito todos los objetivos planteados desde un comienzo.

La **metodología de trabajo** seguida es bastante sencilla y sigue el patrón típico de los que realizan este tipo de trabajos. El autor diariamente trabajaba en las líneas marcadas por los objetivos planteados mediante la investigación y el trabajo autónomo en el laboratorio. Este trabajo implica una gran responsabilidad e iniciativa por parte del autor, ya que es su convicción y ganas de llegar a las metas en este trabajo es lo que hace que la metodología de trabajo haya sido posible y se alcancen, por supuesto, los objetivos marcados.

Adicionalmente, se estipulaban una serie de reuniones semanales con el resto de miembros del proyecto LORO en el propio laboratorio o en alguna sal de reunión de la escuela, entre los que se encontraban la directora de este trabajo Angélica de Antonio Jiménez y la persona que está llevando a cabo su tesis doctoral de la que sale el proyecto y a su vez la doctora Angélica de Antonio también es directora de esta tesis. Gracias a estas reuniones semanales, el autor de este trabajo actualizaba al resto de miembros del grupo sus avances y se le proponían nuevas líneas para encauzar su trabajo o recomendaciones para afrontar diversos problemas y llegar a los objetivos previstos. Todo ello siempre con el apoyo de la directora de este trabajo.

La **planificación** que se ha llevado a cabo en este trabajo ha sido semanal y recogido mediante técnicas de ingeniería del software como son los diagramas de Gantt. Destacar que la planificación original pensada para llegar a los objetivos de este trabajo se ha visto siempre bajo cambios constantes, pero que gracias que se revisaba semanalmente, esto no ha supuesto problema alguno. Esto es debido a que, como el trabajo del autor esta englobado en un proyecto de tesis doctoral, la planificación y algunas de las tareas de los objetivos se han visto modificadas durante el trabajo para adaptarse a las necesidades que se estipulan en la tesis doctoral y el proyecto LORO. Al ser un proyecto de investigación en el área de las EVI, es algo natural y previsible que esto pase. Por ello, la buena planificación y el seguimiento correcto de todo el proyecto ha tenido que ser muy exhaustivo para poder conocer en todo momento que es lo que se tiene por hacer, lo que se está haciendo en curso y lo que ya se ha realizado en el trabajo.

Como se ha mencionado antes, la planificación y seguimiento que se ha realizado enteramente por el autor se encuentra en el diagrama de Gannt que se detalla más abajo (ver FIGURA 18), el cual se actualizaba de forma semanal como ya se ha dicho con los

objetivos realizados en esa semana en el color azul y los planificados desde un primer momento al comienzo del trabajo en el color rojo. Como se puede observar también, se desglosan las 16 semanas de las que se ha compuesto el trabajo y que todos los objetivos que aparecen en el diagrama se corresponden enteramente con los que se han detallado en el capítulo primero en la parte de los objetivos planteados.

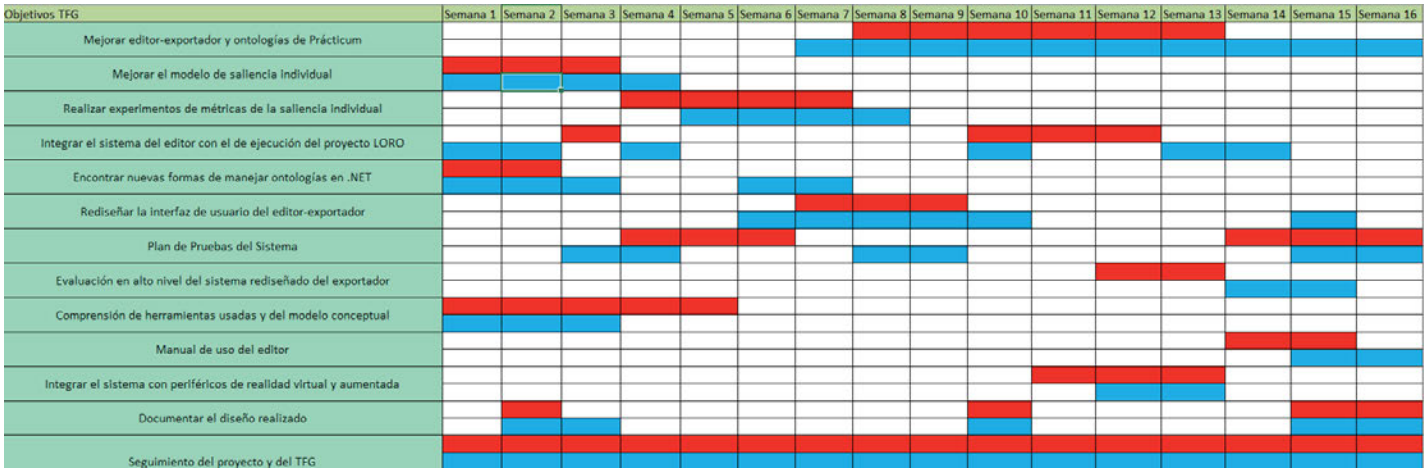


FIGURA 18. Diagrama de Gantt con la planificación y seguimiento del trabajo

2.4. DESARROLLO DEL TRABAJO

Este trabajo se puede dividir en cinco partes bien diferenciadas que cada una por separado han conseguido alcanzar los objetivos planteados en esta memoria en su primer capítulo: diseño del editor-exportador y las ontologías, desarrollo API de lectura y escritura de ontologías en C#, integración de la aplicación preliminar del proyecto LORO con Oculus Rift, diseño de la interfaz de integración y *login* del proyecto LORO y documentación y manuales.

2.4.1. DISEÑO DEL EDITOR-EXPORTADOR

Aquí se va a tratar de desarrollar el trabajo de mayor magnitud de toda la memoria y que recoge todo lo redactado del marco conceptual y las herramientas de este capítulo para que sea posible.

Se van a tratar dos aspectos en este apartado: el diseño del editor-exportador y el tratamiento de los archivos OBJ de archivos de modelos 3D en Unity 3D.

Primeramente, hablar sobre el **tratamiento de los OBJ** en Unity 3D. Esto fue desarrollado en el trabajo debido a que el proceso de uso de los vectores de Zernike (ver Pág. 24) requería de un script que exportara cualquier modelo que hubiera en la escena a un archivo OBJ que es lo que maneja el programa de Binvox que genera los descriptores de Zernike.

Por ello en este trabajo se ha desarrollado un script que recoge toda la malla de los objetos y sus *mesh* de Unity 3D (que es algo así como su relleno), y los traduce a un archivo OBJ para que este programa externo los pueda reconocer y llevar los descriptores de Zernike que genera a la información de cada objeto en las ontologías, a través del editor-exportador.

Seguidamente, hablar del editor-exportador.

El **editor-exportador** es una aplicación realizada sobre el EditorWindow de Unity 3D que permite crear tus propias ventanas para el editor de Unity. Aprovechando esto, el autor ha desarrollado un front-end de la aplicación que va sobre esta facilidad de Unity y todo el back-end que se encarga de recoger la información de estas ventanas personalizadas y llevarlo a la ontología específica seleccionada o crear una nueva a partir de la de modelo.

La misión principal del editor-exportador es ya bien sabida, trata de recoger la información de cada uno de los objetos del escenario y de forma automática llevar esa información a la ontología correspondiente. La información que maneja es muy variada sobre los objetos, pero básicamente se pueden reseñar como sigue: la saliencia individual,

parámetros geométricos (alto, largo, ancho, volumen, orientación y posición), colores, propietario, breve descripción, material y descripción en lenguaje natural de la forma del mismo.

Para poder desplegar la ventana del editor-exportador, simplemente hay que pulsar el botón derecho del ratón sobre el objeto que queramos exportar en el *hierachy* de Unity. De entre las opciones del menú desplegable, y si están instalados los scripts del editor-exportador, se mostrará la opción de "Export Object – LORO Project".

Entonces se nos desplegará una ventana donde se nos insta a crear una nueva ontología específica o a modificar alguna existente (ver FIGURA 19). Sea cual sea la que se seleccione, el editor-exportador te lleva a una nueva ventana (ver FIGURA 20) donde se puede comprobar las propiedades calculadas del objeto y confirmar su exportación a la ontología seleccionada. Pero además ofrece muchas más facilidades:

- Creado y borrado de clases relacionados con los objetos (Content Objects o Structural Objects).
- Introducción de nuevas propiedades a los objetos que se vayan a exportar, más allá de las antes mencionadas en párrafos anteriores de este apartado.
- Gestión de las instancias creadas, con la posibilidad de gestionar que objetos hay ya en las ontologías y poder borrarlos.
- Asignación de áreas de conocimiento a los objetos, así como la posibilidad de crear o borrar las áreas de conocimiento ya existentes en la ontología.
- Elementos de ayuda para dar a conocer al usuario lo que ha de hacer o que ha de rellenar en cierto apartado de algunas propiedades.
- Posibilidad de volver hacia atrás para realizar la selección de una nueva ontología.

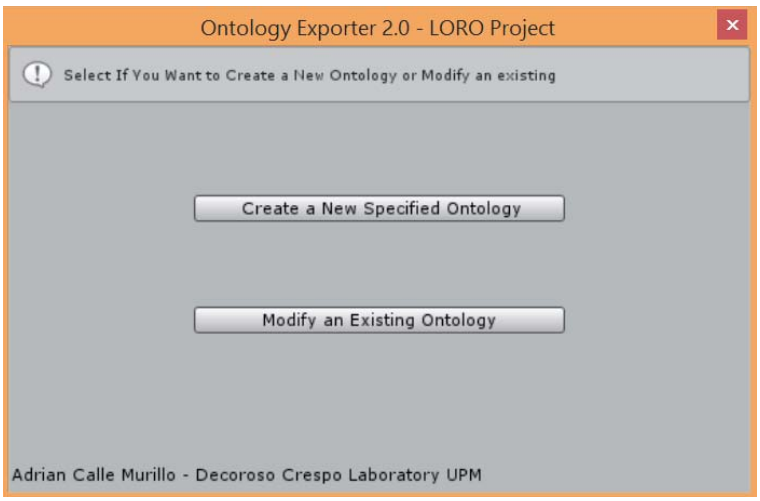


FIGURA 19. Interfaz de inicio del editor-exportador al seleccionar un objeto.

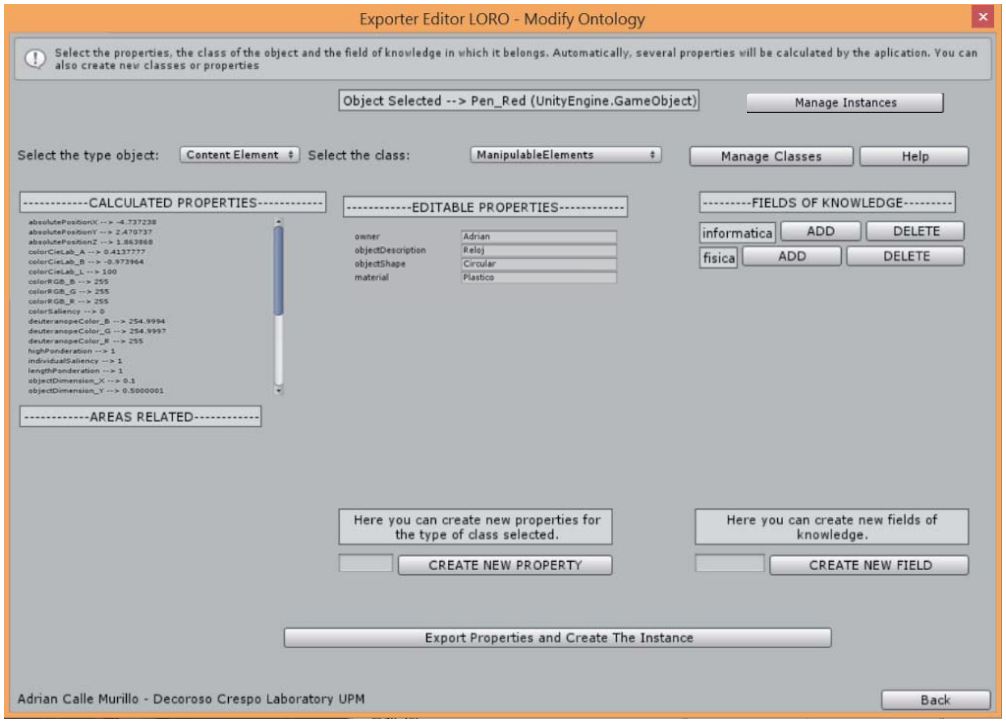


FIGURA 20. Interfaz de editor-exportador.

Con todo ello, de una forma simple y muy intuitiva, cualquiera puede llevar a las ontologías que desee la información de cada uno de los objetos con simplemente tres pulsaciones de ratón. Bien es cierto que esto sería si el usuario sólo quiere llevar el objeto a la ontología sin más, pero como se ha dicho antes, el exportador ofrece multitud de facilidades más para la gestión de al ontología. Igualmente estas opciones extra son usables e intuitivas para el usuario.

Destacar que este exportador, se compone de varios scripts que despliegan tanto el front como el back de la aplicación; y que a su vez, para poder realizar todas las operaciones con las ontologías se comunica con la API que también ha desarrollado el autor para ello y que se detallan en el apartado siguiente (ver Anexo B).

Resaltar también que el editor-exportado basa sus cálculos en el marco conceptual que ya ha sido descrito en este capítulo (véase Pág. 18). Toda la información que se extrae de los objetos sigue el modelo de saliencia individual y el diseño de las ontologías.

Para poder seguir el modelo de saliencia individual, el autor ha creado un algoritmo de ponderaciones muy complejo, el cual se incluye en cualquier parte del exportador que implique crear una nueva instancia de objeto en la ontología o su borrado. Este consiste simplemente en coger para cada una de todas las propiedades que se calculan automáticamente en el exportador y recoger de todas las instancias de objetos que haya en la ontología seleccionada el mayor de todos para cada propiedad. Si el mayor de todos

es el que se va a crear, se actualizan todas las instancias de la ontología con los nuevos valores en base al nuevo objeto que pondera al resto. Si no es así, se modifica solamente el objeto que se vaya a crear en función al mayor que ya pondere al resto en al ontología. Si se borra pasa lo mismo, solo que se comprueba si el que se va a borrar era el ponderador. Si es así, se actualizan todas las instancias de la ontología con los nuevos valores en base al nuevo objeto que se encuentre en la ontología que pondera al resto; si no, no se hace nada (ver FIGURA 21).

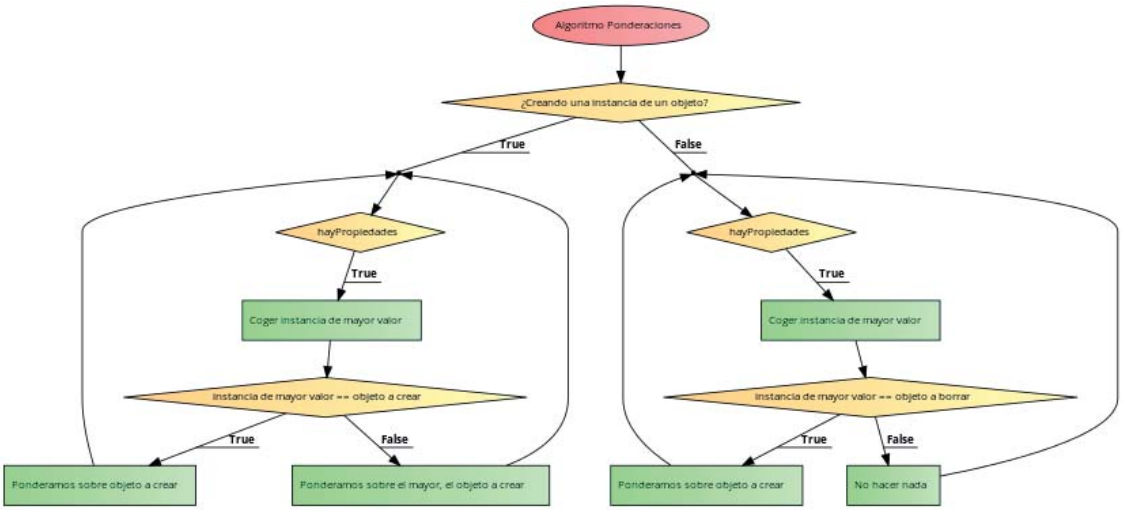


FIGURA 21. Algoritmo de Ponderaciones.

Este editor-exportador ha tenido una versión preliminar que fue realizado durante el periodo de prácticas del autor y que ha servido de base fundamental para el desarrollo de esta versión, mucho más completa y usable. La versión preliminar solo daba la posibilidad de exportar el objeto y no daba las facilidades y tareas extra que se pueden realizar ahora; además, gracias a la rediseño de las ontologías y al uso de las nuevas API, el editor-exportador de este trabajo se ha vuelto más eficiente y preciso en sus cálculo que el de su versión preliminar.

Además este editor-exportador está pensado ya hacia una versión de integración con la segunda parte del proyecto LORO, con lo cual a la interfaz de integración (véase Pág. 46), y hacia lo que podría ser una mejora futura para tener una versión auto contenida y distribuible por la *asset store* de Unity.

Por último decir que en este trabajo el autor también ha realizado tareas de arreglar problemas en las versiones preliminares del editor-exportador y de as ontologías con el objetivo de conseguir una versión estable hasta que se tomó la decisión de rediseñarlo todo de nuevo. Esto se debía a que en el plan de pruebas que también fue realizado en el editor-exportador (véase Pág. 49) se destacaron errores que hubo que corregir para que funcionaran en base a conseguir esta versión estable que más tarde se desecharía.

2.4.2. DESARROLLO API LECTURA Y ESCRITURA ONTOLOGÍAS EN C#

Uno de los aspectos clave para que lo explicado en el apartado anterior tenga sentido y se puedan comunicar entre sí, es el uso del *framework* de Jena y ODNA (véase Pág. 34) para poder hacer que el editor-exportador maneje las ontologías (y también la interfaz de integración que se explica más adelante) y lleve a cabo la misión de leer de ellas y guardar datos en ellas.

Para hacer este cometido mucho más fácil, no sólo para los objetivos de este trabajo por parte del autor sino para el resto de los integrantes del proyecto LORO, el autor desarrolló en lenguaje C# una API que contiene métodos necesarios para que se comunique el exportador y la interfaz de *login* con las ontologías que se creen o manejen en el proyecto.

Gracias a esta API cualquier aplicación en C# puede llamarla y hacer las acciones de forma transparente, incluso si no se tienen ningún tipo de conocimiento sobre ingeniería ontológica. Por ello esta API es importante en el proyecto, ya que cualquier miembro del mismo que necesite realizar operaciones con ontologías lo pueda hacer de la manera más cómoda y legible para llevar a cabo sus propios objetivos.

Estos métodos se destacan en la figura que se encuentra debajo de estas líneas (ver FIGURA 22) con cada uno de los nombres de los métodos, lo que devuelven y lo que reciben en el lenguaje UML.

FIGURA 22. Métodos de la API de comunicación de ontologías sobre Apache Jena.

2.4.3. INTEGRACIÓN DE LA APLICACIÓN EN OCULUS RIFT

Uno de los objetivos que se plantearon en el primer capítulo era tratar de dotar de mayor interacción e inmersión a la potencialmente experiencia de usuario con la aplicación completa del proyecto LORO, no sólo de la primera parte de la que trata este trabajo.

Para ello se decidió emplear el sistema de realidad virtual de Oculus Rift del cual hay una muestra de su hardware en el laboratorio Decoroso Crespo. Varios miembros del proyecto LORO así como por supuesto el autor de este trabajo, se dedicaron durante una semana entera a tratar de montar la aplicación completa del proyecto LORO en sus dos partes para poder tener una versión estable y a su vez integrarlo todo con este sistema de realidad virtual.

El resultado fue que el sistema se integró con éxito en este sistema de Oculus Rift; pero había problemas a la hora de realizar una versión estable del sistema en la segunda parte del proyecto. Esto era debido a que como se emplean más de una cámara en Unity 3D para recoger la información de los usuarios y su percepción, como Oculus sólo usa una cámara, pues entraba en conflicto con las cámaras ya implementadas en el sistema del proyecto LORO.

Por eso una de las líneas de futuro trabajo es mejorar este aspecto pero que en este trabajo ya se ha aproximado bastante, y que servirá poder alcanzar este objetivo con mayor satisfacción (ver Pág. 60).

2.4.4. DISEÑO DE LA INTERFAZ DE INTEGRACIÓN DEL PROYECTO LORO

Uno de los gruesos del trabajo del autor y de los objetivos marcados en el primer capítulo de esta memoria, es la realización de una interfaz en Unity 3D que sirva de puente entre el editor-exportador en tiempo de pre ejecución y la aplicación en sí de la generación de indicaciones del proyecto LORO.

Esta interfaz tiene dos misiones importantes que realizar y son vitales para que el proyecto LORO pueda desarrollarse:

- Recoger la información preliminar de los usuarios y los objetos en función de la escena con la que se quiera trabajar → Esta misión es cargar la ontología del escenario sobre el que se quiera trabajar y que haya sido seleccionado previamente en la interfaz (ver FIGURA 23) y posteriormente cargar toda la información de los objetos que hayan sido exportados a la ontología y de los usuarios que se encuentran en ella para que se pueda decidir en la interfaz con cual trabajar de todos (ver FIGURA 24).

FIGURA 23. Detalle de la selección de ontología y escenario en la interfaz de integración del proyecto LORO sobre Unity 3D.

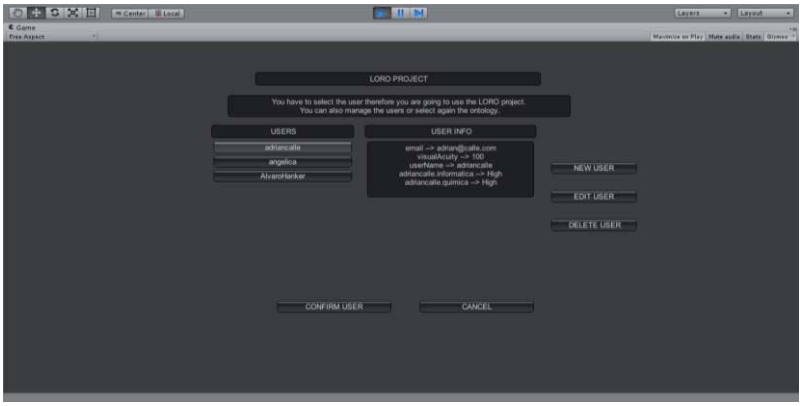


FIGURA 24. Detalle de la selección usuario en la interfaz de integración del proyecto LORO sobre Unity 3D.

- Gestión de los usuarios → La otra misión de la interfaz de integración es dotar la capacidad de poder gestionar los usuarios de la ontología y escena que se hayan cargado en Unity 3D. Para ello, la interfaz ofrece la posibilidad de gestionar los usuarios de la forma de crearlos, editarlos y borrarlos. Cuando se crean o editan usuarios se pueden introducir todos los campos y propiedades mencionadas en el modelo de usuarios de este documento (ver Pág. 31) de forma cómoda y usable para el usuario del sistema (ver FIGURA 25).

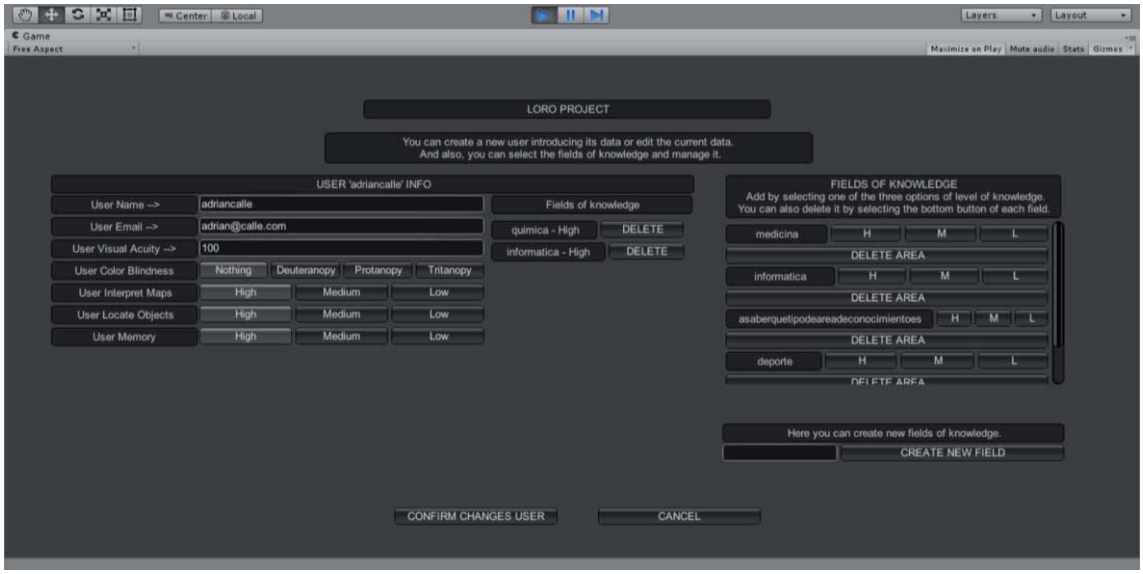


FIGURA 25. Detalle de la edición y creado de usuarios en la interfaz de integración del proyecto LORO sobre Unity 3D.

Hay que destacar que esta interfaz de integración, al igual que el editor-exportador, ha sufrido un rediseño por el cambio de ontologías. Durante este trabajo ya se realizó una versión preliminar de la interfaz de integración para el editor-exportador en su versión también preliminar. Luego esta versión es un resultado de los errores aprendidos y los múltiples cambios en los diseños que han experimentado el exportador y las ontologías a lo largo de todo este trabajo.

2.4.5. DOCUMENTACIÓN Y MANUALES

La realización de esta memoria y de las dos entregas preliminares del Trabajo de Fin de Grado como son el Plan de Trabajo y la Memoria de Seguimiento, se ha realizado durante el desarrollo de este trabajo en las semanas que eran indicadas para poder realizarse y con el contenido exigido mínimo de cada una de ellas.

El **Plan de Trabajo** consiste en una entrega en la segunda semana de trabajo de las 16 de las que se compone éste, y versa únicamente en estipular claramente los objetivos planteados y realizar una planificación a todo el trabajo gracias a un diagrama de Gantt con todas las tareas planificadas. Así mismo, se incluye una copia de la propuesta de trabajo que realizó la directora de este proyecto.

La **Memoria de Seguimiento** es una entrega realizada a mitad del desarrollo del proyecto (hacia la octava semana) con el propósito de dar una visión actualizada de los objetivos planteados originalmente y del diagrama de Gantt con la planificación nueva y el seguimiento realizados hasta entonces. También se incluyen en este documento información relativa a un borrador de dos puntos de esta memoria final como son la introducción y el estado previo del trabajo.

Y por último, esta **Memoria Final** la cual tiene sus detalles de su estructura y lo que contiene en el apartado correspondiente en el primer capítulo de la misma (véase Pág. 13).

Adicionalmente, uno de los objetivos de este trabajo era realizar un completo manual para el usuario y para el experto que hablar sobre el uso del editor-exportador desarrollado y de la interfaz de integración. Esto se hizo con dos objetivos: tener un manual para los miembros del proyecto LORO que haga que puedan usar las aplicaciones desarrolladas sin problema, y otro para tener un manual de los potenciales usuarios de la aplicación en un uso real, ya sea como expertos o como usuarios comunes.

2.5. PLAN DE PRUEBAS DEL SISTEMA

Algunas partes de desarrollo del trabajo ha tenido que ser sometidos a planes de prueba y experimentos para poder determinar que cumplen con las expectativas y objetivos marcados para ellos. Estos son: la métrica de la saliencia individual por forma y el editor-exportador.

El **experimento de la forma**, como se denominan las pruebas diseñadas para la métrica de la saliencia por forma, trata de discernir si la forma de calcular esta métrica (véase Pág. 24) es acorde con lo que piensan los usuarios de ella y lo que perciben realmente de las formas más salientes de los objetos.

Antes de nada, muchos os preguntaréis el por qué no hay experimentos similares para las otras dos métricas por saliencia individual como son el color y el tamaño. La justificación es bastante sencilla. La métrica de color está basada en un artículo (véase Ref. 4) que ya tiene detallados los experimentos que determinan cuales son los colores más salientes, y que por lo tanto ya es una métrica validada y estudiada bien a fondo. Para el caso de la métrica de la saliencia por tamaño, su razón fundamental es el sentido común, Se trata de una métrica con una forma de cálculo muy razonable y que tiene en cuenta todos los parámetros del tamaño posibles, además de haber sido ya experimentada y validada en la tesis doctoral sobre al que se basa este trabajo y el proyecto LORO (véase Ref. 18).

El experimento ha sido diseñado desde cero por el autor y los miembros del proyecto LORO; pero es el auto quien lo ha implementado y realizado en su totalidad sobre la plataforma de Unity 3D al igual que el editor-exportador.

Se apoya en usar una base de datos de modelos 3D en formato OBJ conocida como la *Princenton Shape Benchmarck* (véase Ref. 12), de los cuales el autor ha extraído de forma totalmente aleatoria 100 de estos modelos para realizar el experimento.

La idea del experimento es coger a usuarios sin importar para nada su condición, sexo, edad, capacidades o conocimientos; y mostrarles 25 escenarios con 4 objetos de los 100 que se han seleccionado por el autor (en total harían 100 como se ve), y decir a los usuarios que los ordenen de mayor menor por la saliencia en cuanto a su forma que ellos perciben. Para hacer el experimento lo más justo posible y sacar buenas conclusiones, se muestran los objetos bajo la misma escala de 1 centímetro cúbico y con un color uniforme e igual para todos. Con ello conseguimos que el usuario se fije únicamente en los detalles de la forma y no influya nada más.

La forma de llevar a cabo el experimento diseñado por el autor es muy sencilla. Primeramente, se muestra una interfaz de *login* donde el usuario introduce su nombre,

edad y sexo y comienza el experimento (ver FIGURA 26). A partir de ahí se muestran cada uno de los escenarios con los cuatro objetos para que pueda ordenarlos el usuario.

La forma de ordenar los objetos es mediante su pulsado (mostrando un objeto fantasma de guía al usuario ver FIGURA 27) y colocándolos en la peana que deseen , siendo la más a la izquierda la que registra el objeto más saliente y la que se encuentra a la derecha la menos saliente (ver FIGURA 28).

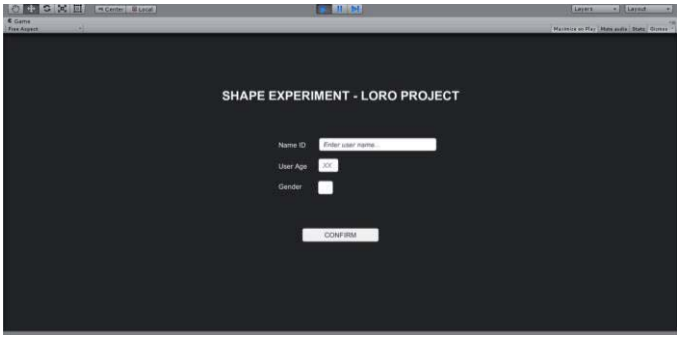


FIGURA 26. Pantalla de inicio de la realización del experimento de la forma.

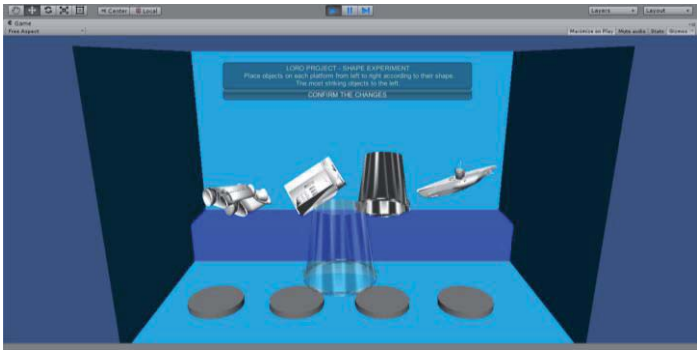


FIGURA 27. Escenario experimento de la forma. Usuario seleccionando y ordenando un objeto colocándolo en la peana.

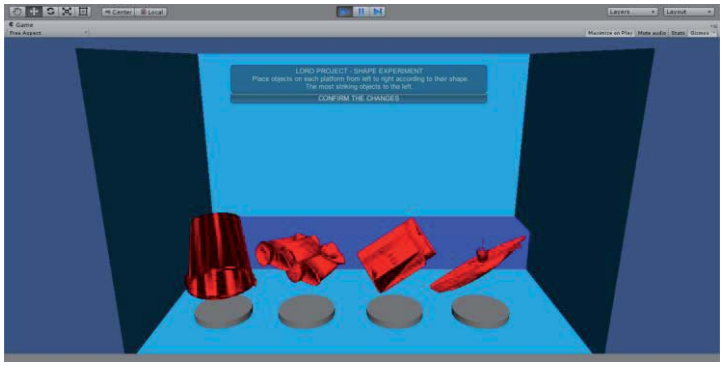


FIGURA 28. Escena con los objetos colocados según saliencia por forma del usuario. El más a la izquierda es el más saliente.

Cuando se haya completado la escena se pasa a la siguiente y se registra toda la actividad del usuario en un fichero CSV (*comma separated value*) donde se registra por escena: orden de los objetos, secuencia de clicks, número de clicks y tiempo de realización. Con este formato se hace muy fácil manejar la información en hojas de cálculo para después sacar las conclusiones de los experimento en el capítulo siguiente de esta memoria.

El **plan de pruebas del editor-exportador** es algo mucho más sencillo.

Se trata de realizar pruebas de integración de todo el exportador con respecto a comprobar la salida de las propiedades calculadas es correcta o no en base a lo esperado, y que ha sido calculado a mano por parte del autor. Este plan de pruebas se ha realizado tanto en la versión preliminar del editor-exportador como en la versión mostrada en este trabajo del exportador.

Para la realización de las pruebas se han empleado tres escenarios diseñados por completo en Unity 3D por miembros del proyecto LORO y del laboratorio Decoroso Crespo que han prestado al autor para hacer sus experimentos. Estos escenarios son los siguientes:

- Escenario Oficina-Diego → Se trata de una sala de oficina de una sola sala y dos mesas con diversos objetos (ver FIGURA 29).

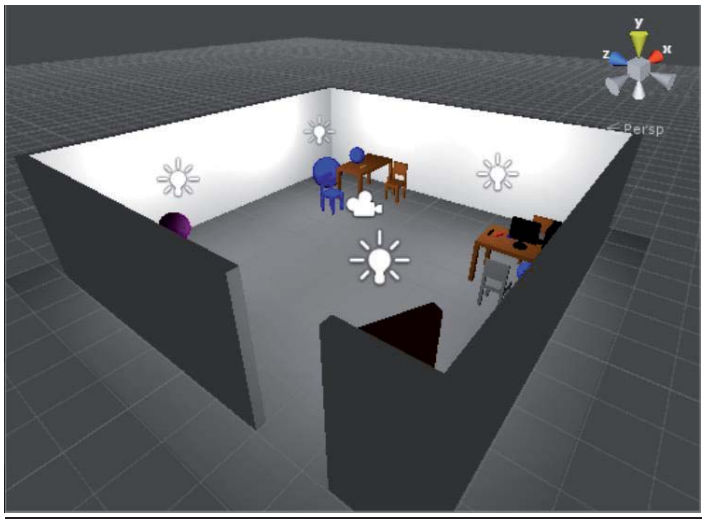


FIGURA 29. Escenario Oficina-Diego.

- Escenario simple → Se trata de un escenario con pocos elementos y compuesto únicamente por una sala (ver FIGURA 30).

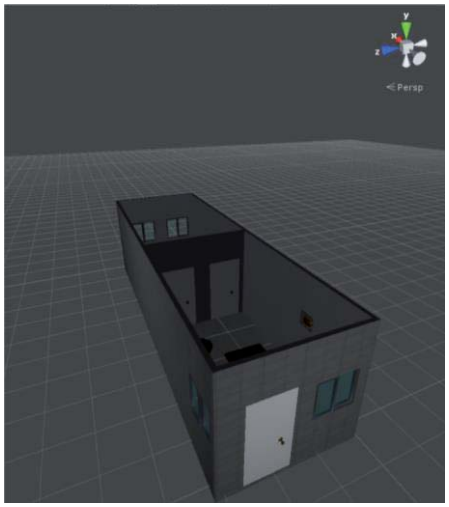


FIGURA 30. Escenario Simple.

- Escenario semi-complejo → Escenario que refleja un edificio de dos plantas con muchos más elementos como un ascensor, pecera, varias salas, objetos diversos,… (ver FIGURA 31).

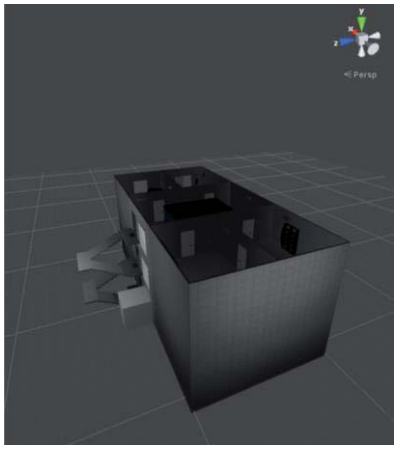
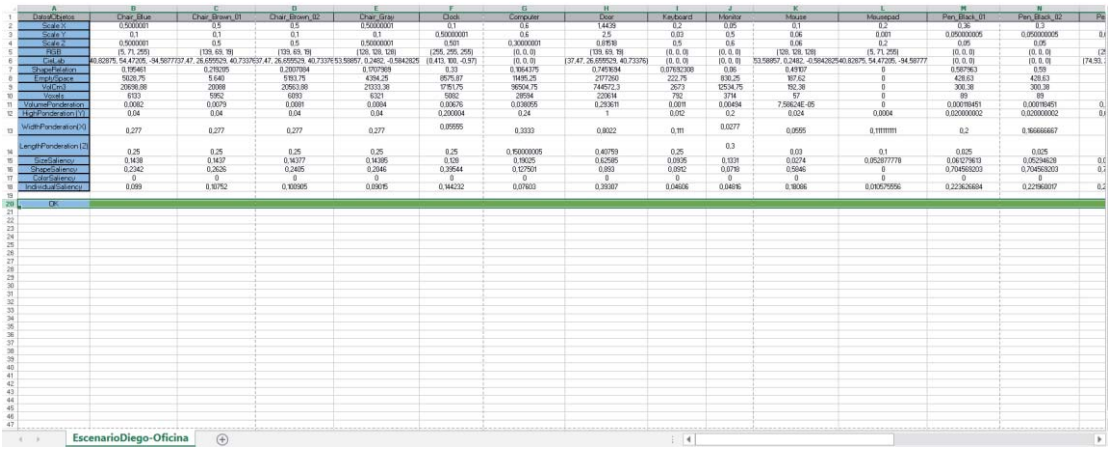


FIGURA 31. Escenario Semi-complejo.

Con cada uno de ellos se recoge la información de tres objetos y se realizan los cálculos a mano por parte del autor para comparar con la salida del algoritmo de ponderaciones del exportador y de los datos de saliencia individual (ver FIGURA 32). El autor guardó esta información en un archivo de hoja de cálculo para facilitar los mismos y la realización de las pruebas.



DatosObjetos	Chair_Blue	Chair_Brown_01	Chair_Brown_02	Chair_Gray	Clock	Computer	Door	Keyboard	Monitor	Mouse	Mousepad	Pen_Black_01	Pen_Black_02
Scale X	0,5000001	0,5	0,5	0,5000001	0,1	0,6	1,4439	0,2	0,05	0,1	0,2	0,36	0,3
Scale Y	0,1	0,1	0,1	0,1	0,5000001	0,6	2,5	0,03	0,5	0,06	0,001	0,0500000	0,0500000
Scale Z	0,5000001	0,5	0,5	0,5000001	0,501	0,3000000	0,8139	0,5	0,6	0,06	0,2	0,05	0,05
RGB	(5, 71, 255)	(139, 69, 19)	(139, 69, 19)	(128, 128, 128)	(255, 255, 255)	(0, 0, 0)	(139, 69, 19)	(0, 0, 0)	(0, 0, 0)	(128, 128, 128)	(5, 71, 255)	(0, 0, 0)	(0, 0, 0)
Cie_Lab	(40,82675, 54,47205, -94,58777)	(37,47, 26,655529, 40,73376)	(37,47, 26,655529, 40,73376)	(53,58857, 0,2462, -0,5842625)	(0,413, 100, -0,97)	(0, 0, 0)	(37,47, 26,655529, 40,73376)	(0, 0, 0)	(0, 0, 0)	(53,58857, 0,2462, -0,5842625)	(40,82675, 54,47205, -94,58777)	(0, 0, 0)	(0, 0, 0)
ShapeRelation	0,195461	0,278295	0,2803894	0,1707909	0,33	0,1664375	0,7418184	0,07932398	0,06	0,49167	0	0,567963	0,59
EmptySpace	5026,75	5 640	5193,75	4394,25	8575,87	11495,25	2177250	222,75	930,25	167,62	0	428,63	428,63
VolDim3	20698,98	20998	20563,88	21233,38	17851,75	98504,75	744572,3	2673	12534,75	762,38	0	300,38	300,38
Voxels	6133	5952	6093	6321	5092	29594	220614	792	3714	57	0	89	89
VolumePonderation	0,0062	0,0079	0,0061	0,0064	0,00076	0,030695	0,293611	0,0011	0,00494	7,59624E-05	0	0,00018451	0,00018451
HighPonderation [Y]	0,04	0,04	0,04	0,04	0,200004	0,24	1	0,012	0,2	0,024	0,0004	0,020000002	0,020000002
WidthPonderation [X]	0,277	0,277	0,277	0,277	0,05995	0,3333	0,8022	0,111	0,0277	0,0555	0,1111111	0,2	0,166666667
LenghtPonderation [Z]	0,25	0,25	0,25	0,25	0,25	0,150000005	0,40759	0,25	0,3	0,03	0,1	0,025	0,025
SizeSaliency	0,1438	0,1437	0,14377	0,14385	0,128	0,19025	0,62585	0,0935	0,1331	0,0274	0,052877778	0,06127613	0,05234628
ShapeSaliency	0,2342	0,2626	0,2405	0,2646	0,39544	0,127501	0,893	0,0992	0,0278	0,5946	0	0,704583203	0,704583203
ColorSaliency	0	0	0	0	0	0	0	0	0	0	0	0	0
IndividualSaliency	0,099	0,10752	0,100305	0,09015	0,144232	0,07603	0,39387	0,04606	0,04016	0,18006	0,010575556	0,223626804	0,221968017
OK													

FIGURA 32. Ejemplo de las pruebas realizadas en los objetos del Escenario Oficina-Diego.

Aunque únicamente se hayan destacado estos dos módulos importantes de pruebas durante este trabajo, también es importante recalcar que se han realizado baterías de pruebas sin documentar sobre el editor-exportador.

Estas pruebas se basan en realizar las pruebas de integración y unitarias pertinentes para todos los módulos y métodos de la API creada para la comunicación con las ontologías y el editor-exportador en si creado en Unity 3D. Pero sin embargo, y como ya se ha resaltado, estas pruebas no se han documentado debidamente como las mostradas anteriormente y solamente constituyen la comprobación del correcto funcionamiento de la API y el exportador en los casos de uso más comunes.

CAPÍTULO 3 - CONCLUSIONES

3.1. RESULTADOS OBTENIDOS

En este apartado, se van a desglosar varias partes de esta memoria en cuanto a los resultados obtenidos: los resultados de las pruebas realizadas en el desarrollo del proyecto y los resultados de si se han alcanzado los objetivos previstos del capítulo primero o no.

A la vista del último apartado del capítulo anterior, las **pruebas realizadas** sobre el editor-exportador y sobre el experimento de la métrica para la saliencia por forma pueden ser los siguientes:

- Pruebas editor-exportador → A la vista de la batería de pruebas realizada, los resultados expuestos en este apartado irán referidos al editor-exportador en su versión de este trabajo y una breve reseña a la versión preliminar.

Destacar primeramente la versión preliminar. Esta versión estaba cargada de errores y de resultados inesperados que propiciaron el arreglo de estos errores en este trabajo (véase Pág. 40). Gracias a la detección de estos errores y a la mejora sustancial de los algoritmos como por ejemplo el de las ponderaciones, la versión del editor-exportador mostrada en este trabajo ha conseguido muchos mejores resultados en las pruebas.

Así pues, la versión del editor-exportador ha conseguido superar las pruebas de sistema planteadas al testeo de la salida del algoritmo de ponderaciones y del resto de propiedades con éxito en todos los objetos de las tres escenas seleccionadas para estos experimentos.

Luego es posible afirmar, que tenemos una versión estable y usable para el proyecto LORO y otras tareas del editor-exportador; tal y como se había planteado en los objetivos de este trabajo y del propio proyecto.

- Experimento de la forma → El experimento se realizó sobre unos 50 usuarios de diferente condición tal y como se estipula en el diseño del experimento. Los resultados fueron bastante positivos.

Para sacar conclusiones de los resultados se han realizado comparaciones de cada uno de los "acomodos" de las escenas de los usuarios, con la colocación que daría nuestra métrica. Simplemente se realizan las distancias de cada uno de los experimentos entre el nuestro y el de cada uno de los usuarios para sacar conclusiones. Estas distancias se estipulan con la saliencia por forma que devuelve la métrica en cada uno de los objetos (ver FIGURA 33).

Nuestra métrica →

Métrica Usuario →

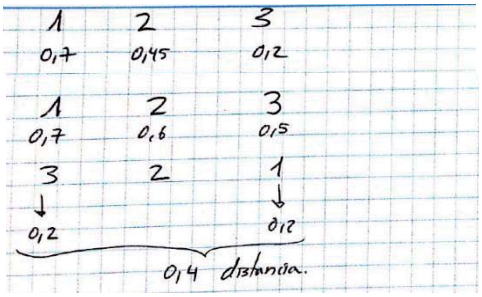


FIGURA 33. Cálculo de distancias de la métrica de la forma en base a nuestro resultado en un experimento.

De estas distancias, que en total son 25 por usuario y por 50 usuarios, extraemos la información relevante. De todas las distancias, más del 65% tienen distancias a nuestra métrica de menos de 0.5 y del 35% restante alrededor del 70% está por debajo de la unidad. La unidad es el valor digamos que si se sobrepasa es el límite admisible para decir que la métrica se adapta al usuario.

Por ello, se puede decir a la vista de los resultados de las distancias obtenidas en este experimento de la forma, que se puede emplear esta forma de medir la saliencia individual por forma de un objeto en base a la percepción visual del objeto por parte de los usuarios.

Destacar por último, que los detalles de estos experimentos se encuentran más detallados en la tesis en curos sobre la que se engloba este trabajo y el proyecto LORO en su totalidad (véase Ref. 18). Y que en ella se detallan muchas más formas de interpretar los resultados del experimento de la forma para discernir si la métrica es buena o no para nuestro modelo de saliencia.

Por ello el autor solamente ha tomado una pequeña referencia de resultados parta este trabajo como parte de la tesis en la que se inscribe el mismo; pero que obviamente, hay muchas más razones que demuestran que la métrica de la saliencia por forma detallada en este trabajo y por el proyecto LORO es válida.

Sobre los objetivos cumplidos, los resultados han sido muy satisfactorios para el autor de este trabajo; ya que la gran mayoría de los objetivos, salvo los dos últimos objetivos, han sido cumplidos bajo las expectativas creadas en un principio para este trabajo.

Bien es cierto que hay que destacar, que de los objetivos que se resalta que han sido cumplidos en este proyecto, hay que decir que algunas de las sub tareas que pertenecen a estos objetivos no hayan sido cumplidas. Para evitar confusiones, en esta memoria se especifica en el segundo capítulo todo lo desarrollado y en este tercer capítulo en el

apartado de líneas futuras de trabajo todas aquellas tareas que no se han cumplido y forman parte de estos trabajos futuros.

Dado que los objetivos se han cumplido en plazo la gran mayoría, lo que cabe destacar ahora es que la planificación y seguimiento han sido buenos a juicio del autor. Aunque a priori pueda parecer lo contrario por lo que refleja el diagrama de Gantt en el capítulo segundo de esta memoria, hay que decir que al estar englobado este trabajo en un proyecto en equipo y una tesis doctoral, lo cambios son admisibles y esperados. Luego los resultados relativos a esta planificación son muy regulares por la falta de previsión, pero el seguimiento semanal ha sido muy bueno ya que se ha controlado gracias a estos diagramas de Gantt en todo momento lo que había que hacer, lo que estaba ya hecho y lo que se estaba haciendo. Facilitando así al autor la realización del trabajo y posteriormente de esta memoria.

Por último, hablar sobre los resultados obtenidos en las aplicaciones y código desarrollados. El autor destaca que el exportador, como se destacará más adelante, es muy mejorable en cuanto su usabilidad, algoritmos para obtener propiedades y para darle la posibilidad de agregar más propiedades. Pero todas estas mejoras estarían fura de lo que es el ámbito del proyecto LORO. Luego, los resultados en este aspecto son muy favorables ya que se han desarrollado aplicaciones funcionales, eficaces y eficientes totalmente para el proyecto LORO y sus objetivos en la primera parte del mismo.

3.2. CONCLUSIONES

La creciente necesidad del uso de los entornos virtuales en muchas de las aplicaciones diarias y de la vida cotidiana, hace que el poder conocer con precisión el entorno sobre el que se trabaja por parte de las máquinas que lo procesan y los propios usuarios que interactúan con ellos sea algo vital y muy importante.

Este trabajo ha conseguido un **avance muy destacable** en este sentido. Ahora los entornos virtuales pueden dotarse de semántica a cada uno de sus objetos, todo ello de forma automática y con las características más importantes de los mismos. Con lo cual, gran parte de los problemas de conocer qué tienen y de qué se componen los entornos virtuales es mucho más sencillo. El trabajo realizado sobre el editor-exportador permite determinar que una silla es una silla, o que una mesa de quirófano es una mesa de quirófano en un entorno virtual; además de agregar las propiedades que son clave en el aspecto del modelo de saliencia individual de cada uno de los objetos para después usarlo en el proyecto LORO y sus posteriores fases de la aplicación.

Otro de los avances de este trabajo están más enfocados al proyecto LORO y se encuentran más acotados a este ámbito para el correcto funcionamiento de la aplicación como la interfaz de integración con la gestión de usuarios, el experimento para la métrica de la saliencia individual de la forma o la integración del sistema preliminar de las dos partes del proyecto en sistemas hardware novedosos de interacción de realidad virtual como es el Oculus Rift.

Por ello, se ha de destacar que la parte más importante de este trabajo el cual ha sido el editor-exportador montado sobre Unity 3D y que se comunica con ontologías, recibe conclusiones por parte del autor de que puede llegar a ser mejorado en sobre manera para aportar mucha más información a los recintos virtuales en futuras mejoras del mismo y también de forma adaptativa y automática. Por ello, sirve de base esencial sobre la que se pueden fijar esas nuevas mejoras para obtener un estandarizado y mucho mejor editor-exportador de cualquier recinto; y quien sabe, la posibilidad de que la repercusión de este editor-exportador trascienda del ámbito más puramente académico y del ámbito de la investigación y su aplicación en los EVI, para pasar a ser una aplicación auto contenida y de distribución en la *Store de Assets* de Unity y tenga una repercusión más comercial si puede decirse, y más popular para su uso a nivel de cualquier entorno virtual.

También cabe la posibilidad de que este editor se pueda distribuir de manera libre y que los usuarios puedan aportar nuevos algoritmos y soluciones para recoger cada vez más datos y características interesantes sobre los objetos en entornos virtuales. O incluso, el poder crear diferentes editores basados en el de este trabajo para poder tener alternativas para recoger información de los recintos en base a las características de éstos.

Resumiendo, la conclusión fundamental es que este avance realizado en este trabajo (refiriéndonos al editor-exportador semántico) es algo que personalmente el autor opina que no ha de quedarse en el olvido y que ha de seguir trabajándose sobre él para poder crear el más completo de los editores, que como ya se ha dicho, es algo crucial en el diseño y conocimiento de los entornos visuales para su posterior procesamiento. Del resto de trabajos realizados, destacar que la conclusión es que se ha cerrado un capítulo muy importante del proyecto LORO, ya que la primera parte del pre procesamiento de los datos para poder llevar a cabo los objetivos del proyecto ya está terminado; y que por tanto, es un gran avance para cerrar el resto de líneas de trabajo que quedan abiertas y finalizar tras mucho tiempo en curso, este proyecto y la tesis doctoral sobre la que se basa.

Para acabar, destacar una pequeña reflexión como conclusión personal del autor sobre este trabajo. Decir simplemente, que ha sido una muy grata experiencia y que el trabajo realizado ha sido muy satisfactorio, habiendo cumplido todos los objetivos marcados desde el principio de esta memoria. Pero sobre todo decir que la finalización de los objetivos era algo que trascendía más allá de este trabajo, ya que el proyecto LORO es un proyecto apasionante y que merece que llegue a buen puerto; y que por tanto, se espera que la aportación del autor al mismo sea muy importante para su finalización.

Además, el ambiente de trabajo en el laboratorio Decoroso Crespo ha sido inmejorable y da una visión completamente diferente a la hora de realizar prácticas a como se nos tienen acostumbrados en la escuela, con una visión más cercana a un mundo laboral y de investigación sobre la que el autor está muy interesado. Cosa que por otra parte es muy importante, ya que el autor ha aprendido gran cantidad de nuevas herramientas y utilidades como pueden ser el manejo de ontologías o el uso de Unity 3D que han hecho enriquecerlo como programador y en su aprendizaje académico.

Luego las conclusiones han sido muy satisfactorias por el trabajo bien realizado y los objetivos previstos cumplidos, y por la grata experiencia personal de haber realizado este Trabajo de Fin de Grado en el mejor de los ambientes y con la mejor gente.

3.3. FUTURAS LÍNEAS DE TRABAJO

A pesar de los grandes avances mostrados, como bien se ha resaltado en el primer capítulo de esta memoria existen aún tareas y trabajos pendientes dentro del proyecto LORO en el que se incluye este trabajo. Recordando que el proyecto LORO se divide en dos partes y que el trabajo detallado en esta memoria se basa en la primera y su parte de integración con la segunda que consiste en el proceso de la generación de indicaciones con los datos de la primera parte, las líneas de trabajo que quedan por desarrollar en el proyecto se encuentran bien marcadas para se puedan completar en un futuro y van fuertemente ligadas con la segunda parte del proyecto y las pruebas de la aplicación completa acabada. Por ello, se destaca lo siguiente:

- Terminar la segunda parte del proyecto, consistente en la recogida de la información de la saliencia individual que prepara este trabajo en las ontologías; y empleando la interfaz de integración también de este trabajo, realizar la generación de indicaciones para un objeto a localizar (LO) en base al objeto de referencia (RO) mejor elegido por su saliencia individual y de contexto.
- Realizar las pruebas referentes a la saliencia de contexto de la segunda parte y a la aplicación completa del proyecto LORO. Para ello se realizarán un total de siete escenarios de diferentes ámbitos para poder llevar a cabo esta tarea y se realizarán las pruebas de forma similar a los experimentos destacados en este trabajo.
- Integrar de mejor forma y en toda su totalidad, la aplicación del proyecto LORO en periféricos de realidad virtual como Oculus Rift o de realidad aumentada, para mejorar la experiencia de usuario y su interacción.
- Mejorar la usabilidad o agregar nuevas funcionalidades al editor-exportador o las ontologías de este trabajo, en tanto sea necesario para el proyecto como para las necesidades del usuario final de la aplicación.
- Integrar reconocimiento de materiales mixtos de color y rediseñado del modelo de saliencia individual en cuanto al color, para adaptarlo a este nuevo cambio que se puede proponer como línea de futura investigación.
- Extrapolar todos los modelos de saliencia y aplicación del proyecto a un ámbito de entornos virtuales de espacios abiertos; además de la posibilidad de integrarlo con sistemas de guiado inteligente en entornos virtuales como los del proyecto MILES que también se ha desarrollado en anteriores trabajos en el laboratorio Decoroso Crespo y supondrían la integración de dos grandes aplicaciones para el navegado y localización de objetos en entornos virtuales de los usuarios.
- Investigar nuevas formas de determinar la saliencia por forma para el modelo de saliencia individual y de contexto; usando de mejor forma los descriptores de Zernike por ejemplo.
- Emplear métodos estadísticos y de clasificación para la decisión del mejor objeto de referencia candidato para las indicaciones de la segunda parte del proyecto en

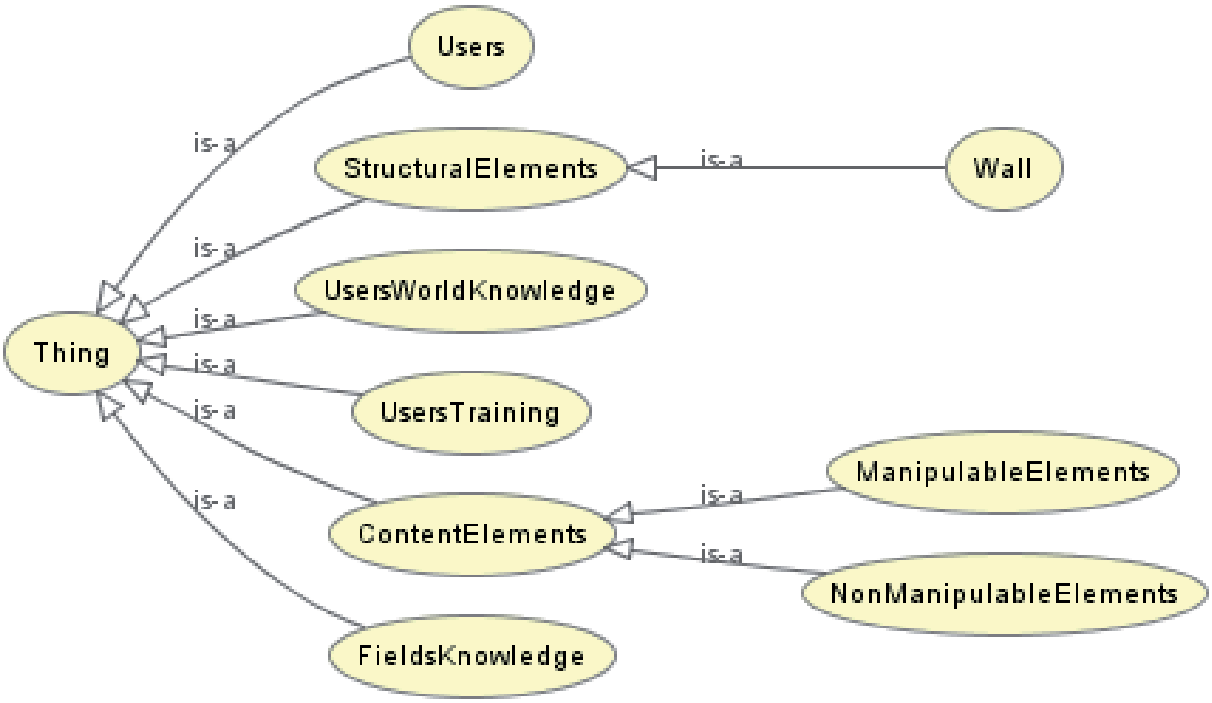
base a su saliencia individual este trabajo; en vez de emplear las ponderaciones de todos los objetos como hasta ahora.

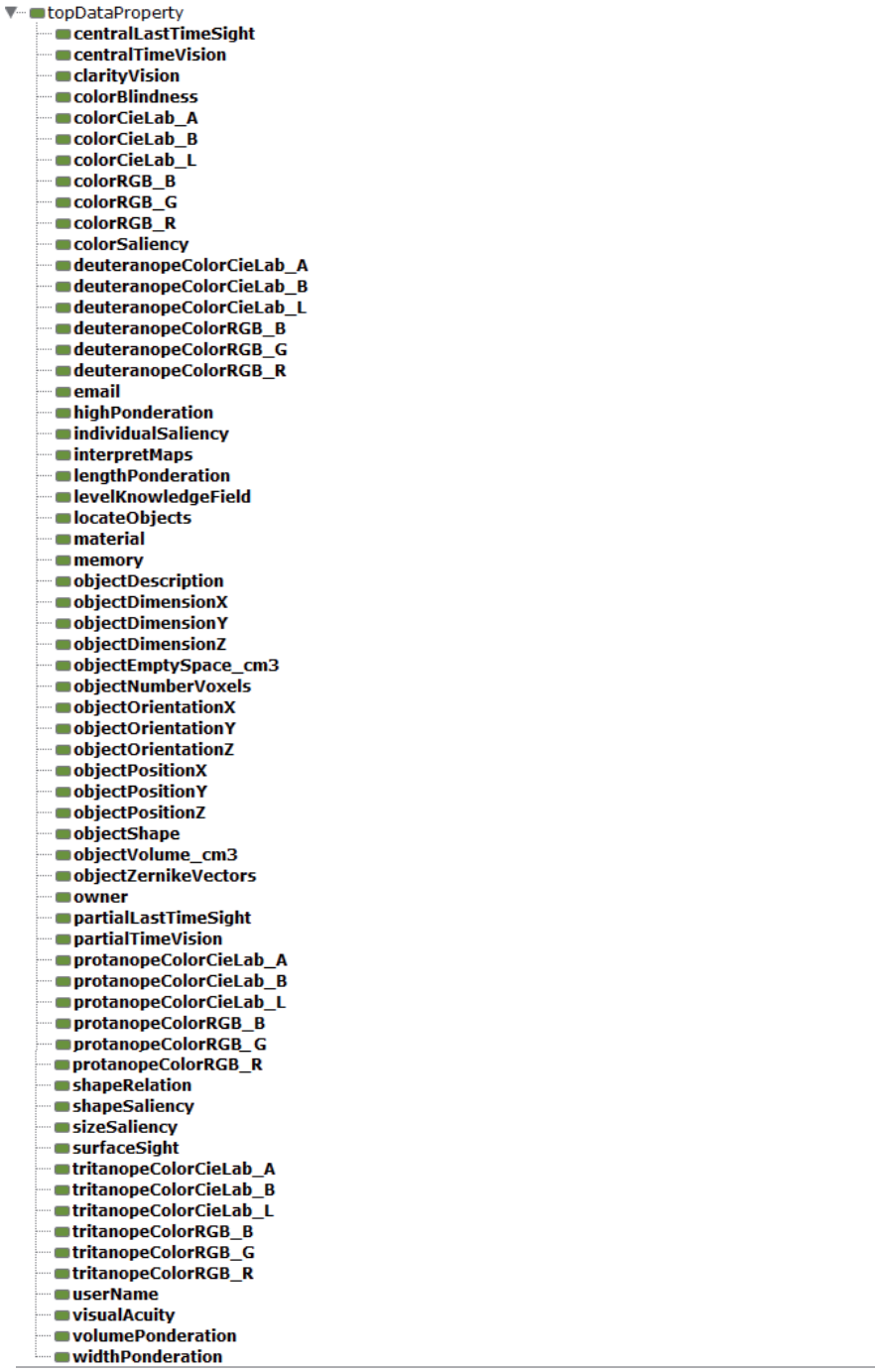
- Mejorar el reconocimiento de objetos según su propósito en el editor-exportador; es decir, detectar por ejemplo que un lapicero es un posible objeto candidato y que por tanto el editor ha de recoger cierta información relevante, o que si es por ejemplo una pared del entorno virtual, pues la trate de manera diferente el exportador.
- Recoger información que se repita de lo usuarios y áreas de conocimiento entre ontologías para evitar trabajo innecesario al editor-exportador.

Bien es cierto que gracias a todo lo que se ha realizado en este trabajo y los trabajos previo sobre el proyecto LORO, se encuentra cada vez más cerca el fin de los objetivos planteados para todo el proyecto y por tanto de la tesis doctoral sobre el que se basa, como ya se ha mencionado en el apartado de conclusiones de este capítulo.

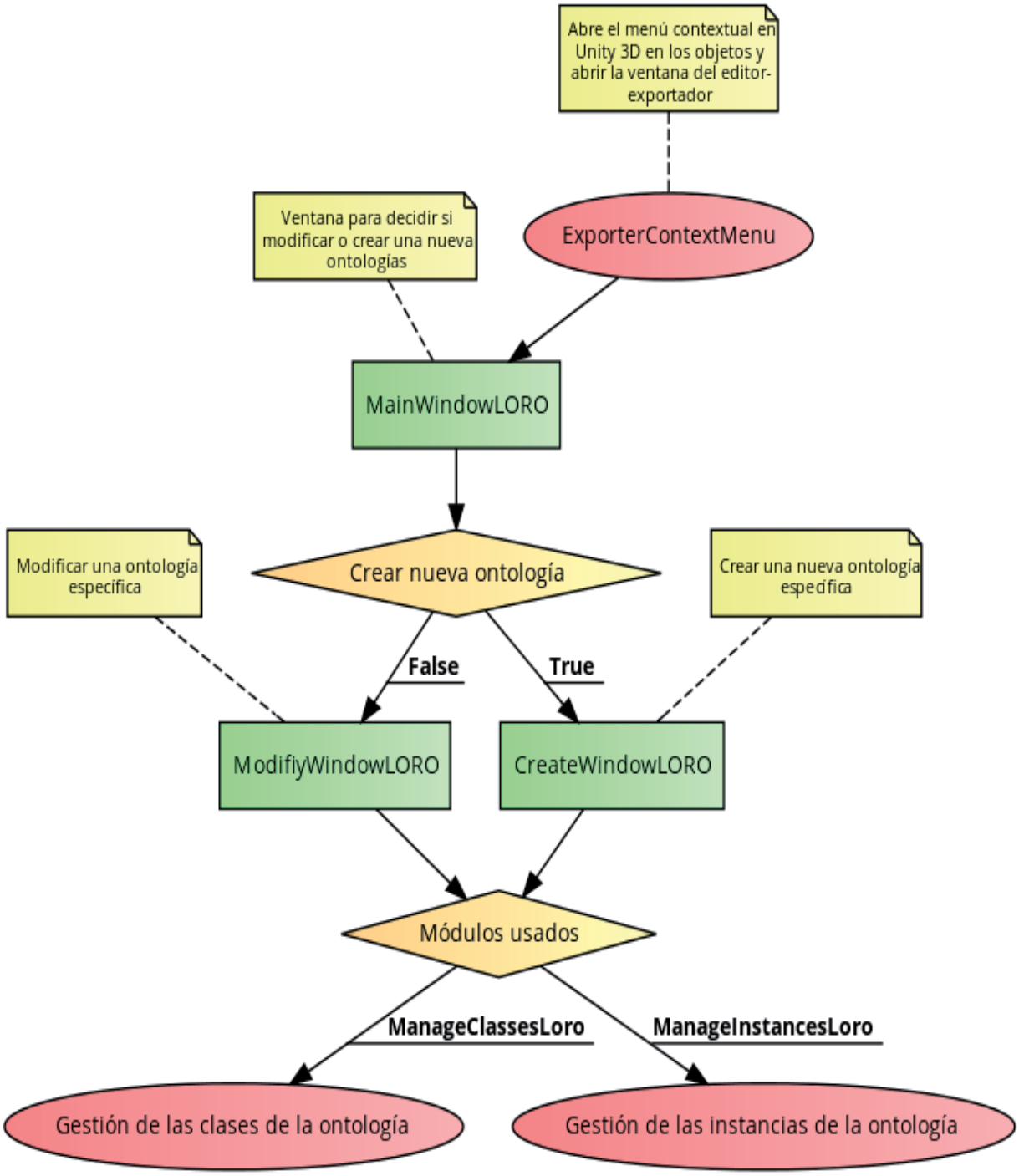
ANEXOS

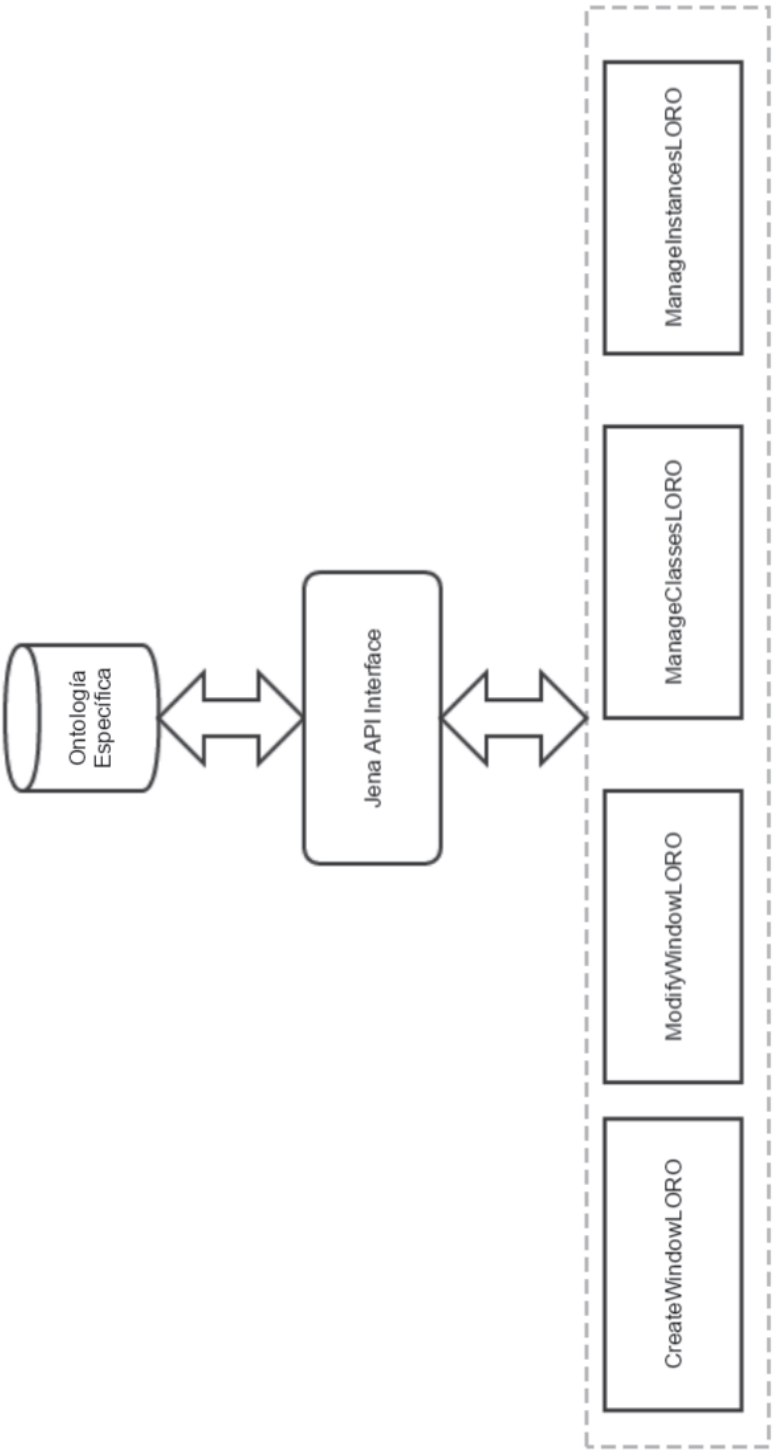
ANEXO A – Clases y Propiedades Ontología Modelo Proyecto LORO.




topDataProperty
centralLastTimeSight
centralTimeVision
clarityVision
colorBlindness
colorCieLab_A
colorCieLab_B
colorCieLab_L
colorRGB_B
colorRGB_G
colorRGB_R
colorSaliency
deuteranopeColorCieLab_A
deuteranopeColorCieLab_B
deuteranopeColorCieLab_L
deuteranopeColorRGB_B
deuteranopeColorRGB_G
deuteranopeColorRGB_R
email
highPonderation
individualSaliency
interpretMaps
lengthPonderation
levelKnowledgeField
locateObjects
material
memory
objectDescription
objectDimensionX
objectDimensionY
objectDimensionZ
objectEmptySpace_cm3
objectNumberVoxels
objectOrientationX
objectOrientationY
objectOrientationZ
objectPositionX
objectPositionY
objectPositionZ
objectShape
objectVolume_cm3
objectZernikeVectors
owner
partialLastTimeSight
partialTimeVision
protanopeColorCieLab_A
protanopeColorCieLab_B
protanopeColorCieLab_L
protanopeColorRGB_B
protanopeColorRGB_G
protanopeColorRGB_R
shapeRelation
shapeSaliency
sizeSaliency
surfaceSight
tritanopeColorCieLab_A
tritanopeColorCieLab_B
tritanopeColorCieLab_L
tritanopeColorRGB_B
tritanopeColorRGB_G
tritanopeColorRGB_R
userName
visualAcuity
volumePonderation
widthPonderation

ANEXO B – Diagrama de Interacción Editor-Exportador.




Ontología Específica
Jena API Interface
ManageInstancesLORO
ManageClassesLORO
ModifyWindowLORO
CreateWindowLORO

BIBLIOGRAFÍA

[1] K. P. Gapp. "Object Localization: Selection of Optimal Reference Objects". *Universität des Saarlandes. Report: SFB 314. Künstliche Intelligenz - Wissensbasierte Systeme*. Bericht NR. 119. 1995. pp. 1-18.

[2] D. D. Hoffman, M. Singh. "Salience of visual parts". *Elsevier Science*. 1996. pp. 29-78.

[3] L. Itti, C. Koch, E. Niebur. "A Model of Saliency-Based Visual Attention for Rapid Scence Analysis". *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Vol. 20. No. 11. 1998. pp. 1254-1259.

[4] A. Choungourian. "Color Preferences and Cultural Variation". *Perceptual and Motor Skills*. 26. 1968. pp. 1203-1206.

[5] P. Doliotis, G. Tsekouras, C.N. Anagnostopoulos, V. Athitsos. "Intelligent Modification of Colors in Digitized Paintings for Enhancing the Visual Perception of Color-blind Viewers". *IFIP International Federation for Information Processing, Volume 296; Artificial Intelligence Applications and Innovations III; Eds. Iliadis, L., Vlahavas, I., Bramer, M.; (Boston: Springer).* 293 – 301. 2009.

[6] C. Connolly, T.A. Fliess. "Study of Efficiency and Accuracy in the Transformation from RGB to CIELAB Color Space". *IEEE Transactions on Image Proccessing, 1997, Vol. 6, No. 7.* 1046 – 1048.

[7] M. Novotni, R. Klein. "3D Zernike Descriptors for Content Based Shape Retrieval". ACM. *Proceedings of the eighth ACM symposium on Solid modeling and applications.* ACM Seattle, Washington, USA, 2003, pp 216 – 225.

[8] L. Chen, J.J. McAuley, R.S. Feris, T.S. Caetano, M. Turk. "Shape Classification Through Structured Learning of Matching Measures". *Computer Vision and Pattern Recognition*. IEEE: 365 - 372. 2009.

[9] Nihilist Dev Blogspot (2012). *Voxelization in Unity.* Available: http://nihilistdev.blogspot.it/2012/08/voxelization-in-unity.html and Available: https://github.com/clynamen/unity-voxelization-script. Last visited: 05/06/2015

[10] J. M. Gascueña, P. González, A. Fernández-Caballero. "Ontologías del modelo del alumno y del modelo del dominio en sistemas de aprendizaje adaptativos y colaborativos". Universidad de Castilla-La Mancha. 2005.

[11] B. Chandrasekaran, John R. Josephson, V. Richard Benjamins, "What Are Ontologies, and Why Do We Need Them?," IEEE Intelligent Systems, vol. 14, no. 1, pp. 20-26, January/February, 1999.

[12] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser (2004). *The Princeton Shape BenchMark*. Available: http://shape.cs.princeton.edu/benchmark/. Last visited: 05/06/2015

[13] H. Knublauch, Ray W. Fergerson, N.F. Noy, M.A. Musen. "The Protegé OWL Plugin: An Open Development Enviroment for Semantic Web Applications" in *The Semantic Web – ISWC 2004*. Springer. 2004. ch. User Interfaces and Visualization, pp. 229-244.

[14] N. F. Noy, D. L. McGuinness. *Ontology Development 101: A Guide to Creating Your First Ontology***.** Knowldege Systems Laboratory, Stanford University. Available: http://protege.stanford.edu/publications/ontology_development/ontology101.pdf. Last visited: 05/06/2015

[15] Joe Raff (2012). *Using Java Classes in a .NET project IKVM*. Available: http://joeraff.org/tag/ikvm/. Last visited: 05/06/2015

[16] D. Helgason, N. Francis, J. Ante - Unity Technologies (2015). *Unity Documentation and Manual.* Available: http://docs.unity3d.com/Manual/index.html. Last visited: 05/06/2015

[17] B. Behlendorf, R.T. Fielding, R. Hartill, D. Robinson, C. Skolnick, R. Terbush, R.S. Tahu, A. Wilson (2015). *Apache Jena Java Framework for Semantic Web*. Available: https://jena.apache.org/index.html. Last visited: 05/06/2015

[18] Lara G., de Antonio A. "Guiding users through virtual enviroments using saliency model" (provisional title), unpublished

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
	Fecha/Hora	Fri Jun 05 23:28:31 CEST 2015
	Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
	Numero de Serie	630
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)